



UNIVERSIDADE  
ESTADUAL de LONDRINA

---

SIMONE SAWASAKI TANAKA

**O PODER DA TECNOLOGIA DE *WORKFLOW* E DOS  
MAPAS CONCEITUAIS NO PROCESSO DE ENSINO E  
APRENDIZAGEM DA UML**

---

Londrina  
2011

SIMONE SAWASAKI TANAKA

**O PODER DA TECNOLOGIA DE *WORKFLOW* E DOS MAPAS  
CONCEITUAIS NO PROCESSO DE ENSINO E  
APRENDIZAGEM DA UML**

Dissertação apresentada ao Curso de Mestrado em  
Ciência da Computação, do Departamento de  
Computação da Universidade Estadual de Londrina,  
como requisito parcial para a obtenção do título de  
Mestre.

Orientador: Prof. Dr. Rodolfo Miranda de Barros.

Londrina  
2011

**Catálogo elaborado pela Divisão de Processos Técnicos da Biblioteca Central da  
Universidade Estadual de Londrina.**

**Dados Internacionais de Catalogação-na-Publicação (CIP)**

T161p Tanaka, Simone Sawasaki.  
O poder da tecnologia de *workflow* e dos mapas conceituais no processo de ensino e aprendizagem da UML / Simone Sawasaki Tanaka. – Londrina, 2011.  
116 f. : il.

Orientador: Rodolfo Miranda de Barros.  
Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2011.  
Inclui bibliografia.

1. Engenharia de software – Estudo e ensino – Teses. 2. Fluxo de trabalho – Teses. 3. Software – Desenvolvimento – Teses. 4. UML (Linguagem de modelagem padrão) – Teses. 5. Mapas conceituais – Teses. I. Barros, Rodolfo Miranda de. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 519.68.02

SIMONE SAWASAKI TANAKA

**O PODER DA TECNOLOGIA DE *WORKFLOW* E DOS MAPAS  
CONCEITUAIS NO PROCESSO DE ENSINO E APRENDIZAGEM DA  
UML**

Dissertação apresentada ao Curso de Mestrado em  
Ciência da Computação, do Departamento de  
Computação da Universidade Estadual de Londrina,  
como requisito parcial para a obtenção do título de  
Mestre.

**BANCA EXAMINADORA**

---

Orientador. Prof. Dr. Rodolfo Miranda de Barros  
Universidade Estadual de Londrina - UEL

---

Prof. Dr. Mario Lemes Proença Jr.  
Universidade Estadual de Londrina - UEL

---

Prof. Dr. Jacques Duilio Brancher  
Universidade Estadual de Londrina - UEL

---

Prof. Dr. Lourival A. de Góis  
Universidade Federal Tecnológica do Paraná -  
UFTPR

Londrina, 19 de outubro de 2011.

Dedico esta dissertação a meu marido, meus filhos  
e meus pais.

# **Agradecimentos**

Agradeço a Deus, por estar presente em todos os meus momentos, fazendo com que me sinta segura com sua presença.

Agradeço em especial ao meu orientador, que não mediu esforços para me ajudar e guiar, estando sempre presente nos momentos de dúvidas e insegurança.

Agradeço ao meu marido e aos meus filhos, pela compreensão e paciência ao longo do mestrado e da produção desta dissertação.

Agradeço aos meus pais, pelo apoio.

Agradeço o companheirismo de todos os colegas do mestrado.

TANAKA, Simone Sawasaki. **O poder da tecnologia de *workflow* e dos mapas conceituais no processo de ensino e aprendizagem da UML**. 2011. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Londrina, 2011.

## RESUMO

Esta dissertação apresenta o estudo da aplicação de um fluxo de trabalho (*workflow*) para o ensino e aprendizagem da modelagem de diagramas utilizando a Linguagem de Modelagem Unificada (UML). Para esta investigação, utilizar-se-ão as características e benefícios dos mapas conceituais para auxiliar no entendimento da elaboração do diagrama, que podem ser utilizados tanto no ensino presencial quanto no ensino a distância. As principais contribuições desta pesquisa foram a implementação do *workflow* para ajudar no ensino-aprendizagem dos modelos da UML e a definição do mapa conceitual, que representa a rastreabilidade para modelagem dos diagramas.

**Palavras-chave:** UML. *Workflow*. Mapa-conceitual. Caso de uso. Ensino e aprendizagem.

TANAKA, Simone Sawasaki. **O poder da tecnologia de *workflow* e dos mapas conceituais no processo de ensino e aprendizagem da UML**. 2011. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Londrina, 2011.

## **ABSTRACT**

This dissertation presents the study of the implementation of a workflow for teaching and learning of modeling diagrams using UML. For this study, use will be made of the features and benefits of concept maps to assist in understanding the development of the diagram, which can be used both in classroom teaching and in distance education. The main contributions of this work have been the implementation of workflow to aid in the teaching-learning models of UML, and the definition of the concept map, which represents traceability for modeling diagrams.

**Keywords:** UML. Workflow. Map-concept. Use case. Teaching and learning.



## Lista De Figuras

Figura 2.1:	Exemplo de Mapa Conceitual Hierárquico [25] .....	23
Figura 2.2:	Relação da terminologia básica associada com <i>workflow</i> [34].....	29
Figura 3.1:	Estrutura da Elaboração dos Diagramas da UML.....	39
Figura 3.2:	<i>Workflow</i> para elaboração do Diagrama de Caso de Uso .....	40
Figura 3.3:	<i>Workflow</i> da Atividade Fazer Diagrama de Caso de Uso (Diagrama de Caso de Uso) .....	41
Figura 3.4:	Mapa Conceitual do Diagrama de Caso de Uso .....	48
Figura 3.5:	<i>Workflow</i> do Diagrama de Atividades .....	49
Figura 3.6:	<i>Workflow</i> da Supertarefa Atividade Elaborar o Diagrama de Atividades .....	50
Figura 3.7:	Mapa Conceitual do Diagrama de Atividades .....	55
Figura 3.8:	<i>Workflow</i> do Diagrama de Classe .....	56
Figura 3.9:	<i>Workflow</i> da Atividade Definir Operações (Diagrama de Classe) .....	57
Figura 3.10:	<i>Workflow</i> da Atividade Definir Atributo (Diagrama de Classes) .....	57
Figura 3.11:	Mapa Conceitual do Diagrama de Classe .....	64
Figura 3.12:	<i>Workflow</i> do Diagrama de Sequência .....	65
Figura 3.13:	<i>Workflow</i> da Atividade “Fazer Diagrama de Sequência” (Diagrama de Sequência) .....	66
Figura 3.14:	Mapa Conceitual do Diagrama de Sequência .....	69
Figura 4.1:	Histograma da questão 2 - Diagrama de Caso de Uso.....	73
Figura 4.2:	Histograma da questão 3 – Diagrama de Caso de Uso .....	74
Figura 4.3:	Histograma da questão 5 - Diagrama de Caso de Uso.....	75
Figura 4.4:	Histograma da questão 7- Diagrama de Caso de Uso.....	75
Figura 4.5:	Histograma da questão 2 - Diagrama de Atividades.....	76
Figura 4.6:	Histograma da questão 3 - Diagrama de Atividades.....	77
Figura 4.7:	Histograma da questão 5 - Diagrama de Atividades.....	77
Figura 4.8:	Histograma da questão 7 - Diagrama de Atividades.....	78
Figura 4.9:	Histograma da questão 2 - Diagrama de Classe.....	79
Figura 4.10:	Histograma da questão 3 - Diagrama de Classe.....	80
Figura 4.11:	Histograma da questão 5 - Diagrama de Classe .....	80
Figura 4.12:	Histograma da questão 7 - Diagrama de Classe.....	81
Figura 4.13:	Histograma da questão 2 - Diagrama de Sequência.....	82

Figura 4.14: Histograma da questão 3 - Diagrama de Sequência.....	83
Figura 4.15: Histograma da questão 5 - Diagrama de Sequência.....	83
Figura 4.16: Histograma da questão 7 - Diagrama de Sequência.....	84

# Lista De Quadros

Quadro 1:	Atividade “Levantar Requisitos Funcionais” .....	42
Quadro 2:	Atividade “Localizar Atores” .....	43
Quadro 3:	Atividade “Identificar Caso de Uso” .....	45
Quadro 4:	Fazer o Diagrama de Caso de Uso.....	46
Quadro 5:	Efetuar a Especificação de Caso de Uso.....	47
Quadro 6:	Atividade “Estabelecer o Foco do Diagrama” .....	51
Quadro 7:	Identificar Grupos e/ou responsáveis.....	52
Quadro 8:	Identificar as Atividades .....	53
Quadro 9:	Elaborar o Diagrama de Atividades .....	54
Quadro 10:	Identificar as Classes do Sistema.....	58
Quadro 11:	Definir Atributos.....	59
Quadro 12:	Definir Operação.....	61
Quadro 13:	Estabelecer Relacionamento .....	62
Quadro 14:	Definir a Multiplicidade.....	62
Quadro 15:	Elaborar o Diagrama de Classe.....	63
Quadro 16:	Seleciona o Caso de Uso.....	66
Quadro 17:	Alocar Atores .....	67
Quadro 18:	Alocar os Objetos Relacionados .....	68
Quadro 19:	Fazer o Diagrama de Sequência.....	68

## **Lista de Siglas e Abreviaturas**

EAD	Educação a Distância
IA	Inteligência Artificial
OMT	Object Modeling Language
RUP	Rational Unified Process
SQL	Structured Query Language
UEL	Universidade Estadual de Londrina
UML	Unified Modeling Language
WfMC	Workflow Management Coalition
WfMS	Workflow Management System

# Sumário

<b>1 INTRODUÇÃO .....</b>	<b>14</b>
<b>2 REVISÃO BIBLIOGRÁFICA .....</b>	<b>17</b>
2.1 APRENDIZAGEM SIGNIFICATIVA .....	17
2.2 MAPA CONCEITUAL .....	21
2.3 UNIFIED MODELING LANGUAGE–UML .....	25
2.4 <i>WORKFLOW</i> .....	28
2.4.1 Modelos de <i>Workflow</i> .....	31
Modelos de Casati/Ceri .....	32
Modelo de Gatilhos .....	32
Modelo de Ações.....	34
2.5 <i>INTEGRAÇÃO WORKFLOW</i> E MAPAS CONCEITUAIS.....	34
2.6 TRABALHOS RELACIONADOS .....	35
2.7 CONSIDERAÇÕES FINAIS.....	36
<b>3 IMPLEMENTAÇÃO DO MAPA CONCEITUAL E DO <i>WORKFLOW</i> .....</b>	<b>37</b>
3.1 ESTRUTURA DOS DIAGRAMAS DA UML .....	37
3.2 MAPA CONCEITUAL E <i>WORKFLOW</i> DO DIAGRAMA DE CASO DE USO .....	39
3.3 MAPA CONCEITUAL E <i>WORKFLOW</i> DO DIAGRAMA DE ATIVIDADES .....	48
3.4 MAPA CONCEITUAL E <i>WORKFLOW</i> DO DIAGRAMA DE CLASSE .....	55
3.5 MAPA CONCEITUAL E <i>WORKFLOW</i> DO DIAGRAMA DE SEQUÊNCIA .....	64
3.6 CONSIDERAÇÕES FINAIS.....	69
<b>4 ESTUDO DE CASO .....</b>	<b>70</b>
4.1 RESULTADOS.....	73
4.1.1 Diagrama de Caso de Uso.....	73
4.1.2 Diagrama de Atividades.....	76
4.1.3 Diagrama de Classe.....	78
4.1.4 Diagrama de Sequência.....	81
4.2 CONSIDERAÇÕES FINAIS.....	84

<b>5 CONCLUSÕES E PESQUISAS FUTURAS.....</b>	<b>85</b>
<b>6 REFERÊNCIAS .....</b>	<b>87</b>
<b>7 Trabalhos publicados pela Autora .....</b>	<b>90</b>
<b>APÊNDICES.....</b>	<b>91</b>
<b>Apêndice A .....</b>	<b>92</b>
<b>Apêndice B .....</b>	<b>94</b>
<b>ANEXO .....</b>	<b>110</b>
<b>Anexo .....</b>	<b>111</b>
<b>Especificação de Caso de Uso [20] .....</b>	<b>111</b>

## 1 INTRODUÇÃO

Para desenvolver um software, é necessário obedecer a uma série de normas e diretrizes e respeitar todo um processo de desenvolvimento, para que se tenha um produto de qualidade. A qualidade pode oferecer segurança tanto em relação aos dados, quanto ao seu funcionamento, atendendo totalmente aos requisitos para os quais ele foi concebido.

Dessa forma, criar mecanismos para facilitar o desenvolvimento de software é uma iniciativa que vem sendo adotada por muitas empresas e institutos de pesquisas para aumentar a produtividade e a qualidade dos produtos intermediários e dos produtos finais. Há diversas formas de aumentar a produtividade, sejam elas ligadas ao desenvolvimento, como direcionados à área de análise e projeto.

No início do desenvolvimento orientado a objetos, haviam vários métodos utilizados na análise, cada um com suas características, porém nenhum completo. Para atender a necessidade de uma padronização foi criada a *Unified Modeling Language* (UML), linguagem de modelagem não muito rígida e nem muito engessada, com escopo nem muito estreito e nem muito abrangente, com o objetivo de atender as necessidades do mundo real [1].

A UML é uma Linguagem de Modelagem Unificada, uma notação (principalmente diagramática) para modelagem de sistemas, usando conceitos orientados a objetos [2]. Fornece um conjunto de diagramas composto por elementos e relacionamentos, que permitem criar modelos gerais para um sistema de software [3]. É definido como modelo uma abstração que retrata a essência de um problema ou estrutura complexa, facilitando sua compreensão [4].

Por ser uma linguagem de modelagem mundialmente utilizada, preocupa-se com seu ensino-aprendizagem. O seu conteúdo está inserido nos cursos de computação no âmbito da disciplina de engenharia de software. As aulas de parte dos professores de UML normalmente são expositivas e dialogadas. Os conceitos são apresentados, porém nem sempre há preocupação em relação aos conceitos que devem ser abordados antes de se iniciar o ensino da linguagem propriamente dito, nem como os diagramas se completam. As atividades em sala normalmente são executadas, partindo de um modelo (*script*), que as tornam mecanizadas. Parte dos alunos se deparam com a dificuldade de visualizar a dependência entre os diagramas e tampouco conseguem obter deles uma visão geral.

A motivação para o desenvolvimento desta pesquisa ocorreu devido as dificuldades apresentadas pelos alunos na disciplina incentivando a busca de uma maneira para facilitar o processo de ensino e aprendizagem. Para tanto, foi proposta a utilização do mapa conceitual visando a melhor entendimento da relação dos diagramas da UML.

Para visualização dinâmica das atividades referentes ao ensino e aprendizagem da UML, adotou-se a tecnologia de *workflow*. A união do *workflow* e dos mapas conceituais permitiu um melhor entendimento na forma de elaborar os diagramas. Por meio de um estudo de caso, foi aplicado o instrumento em sala de aula e analisado o aproveitamento dos alunos, com base em dados obtidos, por aplicação de questionário. A metodologia utilizada para a realização do estudo de caso, bem como o questionário, as respectivas respostas e a análise estatística serão detalhadas no Capítulo 4.

Visando atender ao objetivo traçado por esta pesquisa, foram escolhidos quatro diagramas, baseados na experiência didática de vários professores, de entrevistas informais com alunos e ex-alunos da disciplina e por consultas à literatura que trata dos diagramas essenciais para o desenvolvimento de um sistema, como definido por Lima [17].

Vale ressaltar que os conceitos que fazem parte dos mapas conceituais desenvolvidas no escopo desta pesquisa possuem uma descrição e são acompanhadas de tabelas para ajudar a compreensão do aluno conforme apresentadas no Apêndice A. Algumas delas, além da descrição e gráficos, também trazem a imagem da notação para contribuir para o aprendizado do aluno. Buscando dar suporte ao professor no processo ensino e aprendizagem dos diagramas, este trabalho de pesquisa apresentou como resultados:

- a definição do processo *workflow* dos diagramas propostos (Capítulo 3);
- a instrução de trabalho em relação a cada atividade que compõe o *workflow* (Capítulo 3);
- a criação dos mapas conceituais dos diagramas propostos (Capítulo 3);
- material de apoio em sala de aula (Slides) disponível no endereço [http://www.gaia.uel.br/oa\\_uml/Index.html](http://www.gaia.uel.br/oa_uml/Index.html)
- objeto de aprendizagem disponível no endereço [http://www.gaia.uel.br/oa\\_uml/Index.html](http://www.gaia.uel.br/oa_uml/Index.html)



A partir desta introdução, a dissertação apresenta, no Capítulo 2, os aportes teóricos que são utilizados para sustentação da pesquisa e trabalhos relacionados. O Capítulo 3 enfoca a implementação do Mapa Conceitual dos diagramas da UML e do *Workflow*. O Capítulo 4 aborda o Estudo de Caso com a aplicação do instrumento e respectivos resultados obtidos. Finalmente, no Capítulo 5, apresentam-se as conclusões e considerações para pesquisas futuras.

## 2 REVISÃO BIBLIOGRÁFICA

São apresentados neste capítulo os principais conceitos utilizados nesta dissertação, os quais serviram como base à elaboração do *workflow* e dos mapas conceituais.

### 2.1 APRENDIZAGEM SIGNIFICATIVA

Para Ausubel [5], a aprendizagem pode se processar entre os extremos da aprendizagem mecânica e a aprendizagem significativa. A aprendizagem mecânica está relacionada com a aprendizagem de novas informações, com pouca ou nenhuma associação com conceitos relevantes existentes na estrutura cognitiva do aluno. O aluno simplesmente recebe a informação e a armazena, de forma que ela permanece disponível por certo intervalo de tempo. Na ausência de outras informações que lhe sirvam de combinação, permanece na estrutura cognitiva de forma estática. Este tipo de aprendizado ocorre quando as novas informações são aprendidas, sem interagirem com conceitos relevantes existentes na estrutura cognitiva.

A Aprendizagem Mecânica, segundo Ausubel [5], é necessária e inevitável, no caso de conceitos inteiramente novos para o aprendiz. Todavia, posteriormente, ela tornar-se-á significativa. Para acelerar esse processo, Ausubel propõe os organizadores prévios, âncoras criadas a fim de manipular a estrutura cognitiva, interligando conceitos aparentemente não relacionáveis por meio da abstração.

Dessa forma, a Aprendizagem Significativa é preferível à Aprendizagem Mecânica, pois constitui um método mais simples, prático e eficiente. Muitas vezes, um indivíduo pode aprender algo mecanicamente e só mais tarde vir a perceber que este se relaciona a algum conhecimento anterior já dominado. Nesse caso, teriam ocorrido esforço e tempo demasiados para assimilar conceitos que seriam mais facilmente compreendidos, se encontrasse uma "âncora", ou um conceito subsunçor, existente na Estrutura Cognitiva [24].

Moreira [22] assevera que a aprendizagem significativa é o processo através do qual nova informação (um novo conhecimento) se relaciona de maneira não

arbitrária e substantiva (não-literal) à estrutura cognitiva do aprendiz. É no curso da aprendizagem significativa que o significado lógico do material de aprendizagem se transforma em significado psicológico para o sujeito. É o mecanismo humano, por excelência, para adquirir e armazenar a vasta quantidade de ideias e informações representadas em qualquer campo de conhecimento.

Não-arbitrariedade e substantividade são as características básicas da aprendizagem significativa. Não-arbitrariedade quer dizer que o material potencialmente significativo se relaciona de maneira não-arbitrária com o conhecimento já existente na estrutura cognitiva do aprendiz. Dito de outro modo, o relacionamento não é com qualquer aspecto da estrutura cognitiva, mas, sim, com conhecimentos especificamente relevantes, os quais Ausubel denomina subsunçores. O conhecimento prévio serve de matriz ideacional e organizacional para a incorporação, compreensão e fixação de novos conhecimentos, quando estes “se ancoram” em conhecimentos especificamente relevantes (subsunçores) preexistentes na estrutura cognitiva [22].

A Aprendizagem Significativa, que tem como base as informações já existentes na estrutura cognitiva, é considerada por Ausubel ideia-âncora ou subsunçor. O subsunçor é uma estrutura específica por meio da qual uma nova informação pode se integrar à mente humana. Esta é altamente organizada e detentora de uma hierarquia conceitual que armazena experiências prévias do aluno.

Sendo assim, as novas informações podem interagir e contribuir para transformação do conhecimento em novos conhecimentos, de forma dinâmica, porém relacionada entre a nova informação e os aspectos relevantes da estrutura cognitiva do indivíduo.

Conforme Barbosa [6], a essência do trabalho de Ausubel é a aprendizagem significativa na qual os conceitos são ordenados progressivamente, de forma que os conceitos mais gerais de um conteúdo estão ligados a conceitos subordinados, e estes, a conceitos específicos.

Para Moreira [7], Ausubel sustenta que o ponto de vista de que cada disciplina acadêmica tem uma estrutura articulada e hierarquicamente organizada de conceitos que constitui o sistema de informações dessa disciplina. Acredita-se que esses conceitos estruturais possam ser identificados e ensinados a um aluno, constituindo para ele um sistema de processamento de informações, um verdadeiro mapa intelectual que pode ser usado para analisar o domínio particular da disciplina e nela resolver problemas.

Para que uma aprendizagem ocorra, ela deve ser significativa, o que exige que seja vista como a compreensão de significados, relacionando-se às experiências anteriores e vivências pessoais dos alunos, permitindo a formulação de problemas de algum modo desafiantes que incentivem a expansão da aprendizagem, o estabelecimento de diferentes tipos de relações entre fatos, objetos, acontecimentos, noções e conceitos, desencadeando modificações de comportamentos e contribuindo para a utilização do que é aprendido em diferentes situações.

Isso não significa que todas as noções e conceitos que os alunos aprendem devam estar ligados à sua realidade imediata, o que seria olhar para os conteúdos escolares de maneira muito simplista. Queremos, sim, afirmar que os conteúdos que a escola veicula devem servir para desenvolver novas formas de compreender e interpretar a realidade, questionar, discordar, propor soluções, formar leitores reflexivos acerca do mundo que os rodeia [23].

Falar em aprendizagem significativa é assumir que aprender possui um caráter dinâmico que exige ações de ensino direcionadas para que os alunos aprofundem e ampliem os significados elaborados, mediante suas participações nas atividades de ensino e aprendizagem. Nessa concepção, o ensino é um conjunto de atividades sistemáticas, cuidadosamente planejadas, em torno das quais conteúdo e forma articulam-se inevitavelmente, e nas quais o professor e o aluno compartilham parcelas cada vez maiores de significados com relação aos conteúdos do currículo escolar. Isso significa que o professor guia suas ações para que o aluno participe de tarefas e atividades que o façam se aproximar cada vez mais dos conteúdos que a escola tem para lhe ensinar [23].

Além de saber o que é aprendizagem significativa, conhecemos princípios programáticos facilitadores - como a diferenciação progressiva, a reconciliação integradora, a organização sequencial e a consolidação [5] - e algumas estratégias facilitadoras - como os organizadores prévios, os mapas conceituais e os diagramas V [24].

A diferenciação progressiva, aponta Valério [24], é o princípio programático segundo o qual as ideias mais gerais e inclusivas da matéria de ensino devem ser apresentadas desde o início da instrução e, progressivamente, diferenciadas em termos de detalhes e especificidade. Não se trata de um enfoque dedutivo, mas de abordagem na qual o que é mais relevante deve ser introduzido desde o início e, logo em seguida, trabalhado por meio de exemplos, situações, exercícios. As ideias gerais e inclusivas devem ser retomadas periodicamente, favorecendo assim sua progressiva diferenciação. É um princípio compatível com a progressividade da aprendizagem significativa.

Por outro lado, a programação da matéria de ensino deve não apenas proporcionar a diferenciação progressiva, mas também explorar, explicitamente, relações entre conceitos e proposições, chamar a atenção para diferenças e semelhanças e reconciliar inconsistências reais e aparentes. É nisso que consiste a reconciliação integradora, ou integrativa, como princípio programático de um ensino que visa à aprendizagem significativa. A organização sequencial, como princípio a ser observado na programação do conteúdo com fins instrucionais, consiste em sequenciar os tópicos, ou unidades de estudo, de maneira tão coerente quando possível (observados os princípios de diferenciação progressiva e reconciliação integrativa) com as relações de dependência naturalmente existentes entre eles na matéria de ensino.

A consolidação como quarto princípio programático de um ensino objetivando a aprendizagem significativa leva o domínio (respeitada a progressividade da aprendizagem significativa) do que está sendo estudado antes de se introduzirem novos conhecimentos. É uma decorrência natural da premissa de que o conhecimento prévio é a variável que mais influencia a aprendizagem subsequente.

Organizadores prévios são materiais introdutórios apresentados antes do material de aprendizagem em si mesmo, em um nível mais alto de abstração, generalidade e inclusividade, para servir de ponte entre o que o aprendiz já sabe e o que deveria saber, a fim de que esse material seja potencialmente significativo ou, mais importante, para mostrar a possibilidade de relacionar o novo conhecimento com o conhecimento prévio.

Vários materiais de aprendizagem podem ser utilizados afim de facilitar o processo de aprendizagem de forma significativa, como os mapas conceituais e diagramas V.

Os diagramas V são instrumentos heurísticos para a análise da estrutura do processo de produção de conhecimento (entendido como as partes desse processo e a maneira como se relacionam) e para "desempacotar" conhecimentos documentados sob a forma de artigos de pesquisa, livros, ensaios etc. Assim como no caso dos mapas conceituais, sua construção, discussão e reconstrução são processos bastante favorecedores de aprendizagens significativas.

Já os mapas conceituais são diagramas que indicam relações entre conceitos (apenas conceitos) e procuram refletir a estrutura conceitual de um certo conhecimento. Mais especificamente, podem ser vistos como diagramas conceituais hierárquicos. Construí-los, negociá-los, apresentá-los, refazê-los, são processos altamente facilitadores de uma aprendizagem significativa.

## 2.2 MAPA CONCEITUAL

Mapas conceituais, ou mapas de conceitos, são diagramas indicando relações entre conceitos ou entre palavras que usamos para representar conceitos. Embora normalmente tenham organização hierárquica e, muitas vezes, incluam setas, tais diagramas não devem ser confundidos com organogramas ou diagramas de fluxo, pois não implicam sequência, temporalidade ou direcionalidade, nem hierarquias organizacionais ou de poder [21].

Os mapas conceituais são diagramas de significados, de relações significativas, de hierarquias conceituais, se for o caso. Isso também os diferencia das redes semânticas que não necessariamente se organizam por níveis hierárquicos e não obrigatoriamente incluem apenas conceitos. Mapas conceituais também não devem ser confundidos com mapas mentais que são associacionistas, não se ocupam de relações entre conceitos, incluem coisas que não são conceitos e não estão organizados hierarquicamente. Não devem, igualmente, ser confundidos com quadros sinópticos, que são diagramas classificatórios. Mapas conceituais buscam relacionar e hierarquizar conceitos [21].

Barros [10] afirma que os recursos esquemáticos dos mapas conceituais, que representam um conjunto de conceitos inter-relacionados numa estrutura hierárquica proposicional servem para tornar claras para professores e alunos as relações entre conceitos de um conteúdo aos quais deve ser dada maior ênfase. Em sua essência, proveem representações gráficas de conceitos em um domínio específico de conhecimento, construídas de tal forma que as interações entre os conceitos são evidentes.

Muitas vezes utilizam-se figuras geométricas - elipses, retângulos, círculos - ao traçar mapas de conceitos, mas tais figuras são, em princípio, irrelevantes. É certo que o uso de figuras pode estar vinculado a determinadas regras como, por exemplo, a de que conceitos mais gerais, mais abrangentes, devem estar dentro de elipses e conceitos bem específicos dentro de retângulos. Em princípio, no entanto, figuras geométricas nada significam em um mapa conceitual. Assim como nada significam o comprimento e a forma das linhas ligando conceitos em um desses diagramas, a menos que estejam acopladas a certas regras. O fato de dois conceitos estarem unidos por uma linha é importante porque

significa que há, no entendimento de quem fez o mapa, uma relação entre esses conceitos, mas o tamanho e a forma dessa linha são, a priori, arbitrários [21].

Não há regras gerais fixas para o traçado de mapas de conceitos. O importante é que o mapa seja um instrumento capaz de evidenciar significados atribuídos a conceitos e relações entre conceitos no contexto de um corpo de conhecimentos, de uma disciplina, de uma matéria de ensino. Uma ou duas palavras-chave escritas sobre essa linha podem ser suficientes para explicitar a natureza dessa relação. Os dois conceitos mais as palavras-chave formam uma proposição e esta evidencia o significado da relação conceitual. Por esta razão, o uso de palavras-chave sobre as linhas conectando conceitos é importante e deve ser incentivado na confecção de mapas conceituais. Mas, esse recurso não os torna autoexplicativos. Mapas conceituais devem ser explicados por quem os faz; ao explicá-lo, a pessoa externaliza significados [21].

Mapas conceituais foram propostos e exemplificados como meios instrucionais que podem ser usados tanto na análise e organização do conteúdo, quanto no ensino e na avaliação da aprendizagem. São recursos flexíveis, dinâmicos, utilizáveis em qualquer sala de aula (ou laboratório), cuja maior vantagem pode estar exatamente no fato de enfatizarem o ensino e a aprendizagem de conceitos, algo que muitas vezes fica perdido em meio a uma grande quantidade de informações e fórmulas [21].

Como uma ferramenta de aprendizagem, o mapa conceitual é útil para o estudante fazer anotações, resolver problemas, planejar o estudo e/ou a redação de grandes relatórios, preparar-se para avaliações e integrar tópicos.

A técnica de construção e a teoria a respeito dos mapas conceituais foram desenvolvidas pelo pesquisador norte-americano Joseph Novak (1998). Ele define mapa conceitual como ferramenta para organizar e representar conhecimento. O mapa conceitual, baseado na teoria da Aprendizagem Significativa, de David Ausubel, é uma representação gráfica em duas dimensões de um conjunto de conceitos construídos de tal forma que as relações entre eles sejam evidentes.

Existem diversas formas de fazer os mapas conceituais, como, por exemplo, o mapa conceitual do tipo sistema: “entrada e saída”, que organiza a informação em um formato que é semelhante ao fluxograma, mas com o acréscimo da imposição das possibilidades “entrada” e “saída”, mapa conceitual tipo “fluxograma”, que organiza a informação de maneira linear. Ele é utilizado para mostrar passo-a-passo determinado procedimento e, normalmente, inclui um ponto inicial e outro ponto final.

Um fluxograma é normalmente usado para melhorar o desempenho de um procedimento e mapa conceitual “hierárquico”, no qual informação é apresentada em ordem descendente de importância. A informação mais importante (inclusiva) é colocada na parte superior. Um mapa hierárquico é usado para nos dizer algo sobre um procedimento. A Figura 2.1 apresenta um exemplo de um mapa conceitual do tipo hierárquico [25].



Figura 2.1 - Exemplo de Mapa Conceitual Hierárquico [25].

Segundo Gaines e Shaw [27] (2004, citados por Barros, 2008), os mapas conceituais podem ser descritos sob diversas perspectivas, conforme o nível de análise considerado:

- **perspectiva abstrata:** os mapas conceituais constituídos por nós ligados por arcos podem ser vistos como hipergrafos ordenados. Cada nó tem um identificador único e um conteúdo, enquanto as



ligações entre os nós podem ser direcionadas ou não direcionadas, representadas visualmente por linhas entre os nós, com ou sem flechas nas extremidades;

- **perspectiva de visualização:** os mapas conceituais podem ser vistos como diagramas, construídos através do uso de signos. Cada tipo de nó pode determinar (ou ser determinado) pela forma, cor externa ou de preenchimento, enquanto as ligações podem ser identificadas pela espessura da linha, cor ou outras formas de representação;
- **perspectiva da conversação:** os mapas conceituais podem ser considerados forma de representação e comunicação do conhecimento através de linguagens visuais, porque estão sujeitos à interpretação por alguma comunidade de referência. Essa interpretação permite o estabelecimento de um paralelo entre a linguagem natural e a linguagem visual - as estruturas gramaticais e suas estruturas adquirem significado, segundo são utilizadas em determinada comunidade.

Como ferramenta de aprendizagem, o mapa conceitual é útil para o estudante para:

- fazer anotações;
- resolver problemas;
- planejar o estudo e/ou a redação de grandes relatórios;
- preparar-se para avaliações;
- integrar tópicos.

Para os professores, os mapas conceituais podem auxiliar em suas tarefas rotineiras, tais como:

- ensino de um novo tópico: na construção de mapas conceituais, os conceitos difíceis são esclarecidos e podem ser arranjados em uma ordem sistemática. O uso de mapas conceituais pode auxiliar os professores a se manterem mais atentos aos conceitos-chaves e relações entre eles. Os mapas podem auxiliá-lo a transferir uma imagem geral e clara dos tópicos e suas relações para seus estudantes. Dessa forma, torna-se mais fácil para o estudante não perder ou não entender qualquer conceito importante;

- reforço da compreensão: o uso dos mapas conceituais reforça a compreensão e aprendizagem por parte dos alunos. Ele permite a visualização dos conceitos-chave e resume suas inter-relações;
- verificação da aprendizagem e identificação de conceitos mal compreendidos: os mapas conceituais também podem auxiliar os professores na avaliação do processo de ensino. Eles podem avaliar o alcance dos objetivos pelos alunos pela identificação dos conceitos mal entendidos e aqueles que estão faltando;
- avaliação: a aprendizagem do aluno (alcance dos objetivos, compreensão dos conceitos e suas interligações, etc.) pode ser testada ou examinada por meio da construção de mapas conceituais.

Segundo Amoretti e Tarouco [11], a representação do conhecimento em rede facilita a apreensão do conhecimento, porque a memória humana reconhece e retém mais rapidamente os exemplares prototípicos, respondendo de maneira mais satisfatória às expectativas de realidade dos leitores e facilitando o processo mental da compreensão. A rede simula aspectos típicos da cognição humana, tendo como característica essencial a flexibilidade na modelagem de fenômenos cognitivos que é a capacidade da rede de sempre completar os conceitos descritos através da associação de novas propriedades aos conceitos básicos.

O mapa conceitual é uma ferramenta que auxilia no ensino e na aprendizagem. Uma ferramenta pedagógica capaz de evidenciar significados, levando os conceitos a se tornarem um conjunto, uma teia que se forma, a partir das relações entre estes conceitos que desenvolvem o cognitivo de quem o utiliza.

### 2.3 UNIFIED MODELING LANGUAGE–UML

É através da linguagem que as pessoas se comunicam e procuram se entender. A UML é uma linguagem, que fornece um vocabulário e regras [3], assim como uma linguagem verbal ou não verbal.

A UML é uma linguagem gráfica para visualizar, especificar, construir e documentar os artefatos de um sistema de software. Por meio de seus diagramas, é possível representar sistema de software sob diversas perspectivas de visualização, facilitando a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um

sistema - gerentes, coordenadores, analistas, desenvolvedores, por apresentar um vocabulário de fácil entendimento[3].

A importância da modelagem para um bom desenvolvimento e entendimento de um sistema de software torna a UML indispensável, proporcionando assim melhor comunicação entre todas as pessoas que estão envolvidas no projeto de desenvolvimento do software.

A UML começou a ser definida a partir de uma tentativa de Rumbaugh e Grady Booch de combinar dois métodos populares de modelagem orientada a objeto: Booch e *Object Modeling Language* (OMT). Mais tarde, Ivar Jacobson, o criador do método Objectory, uniu-se aos dois, para a concepção da primeira versão da linguagem *Unified Modeling Language* (UML).

Os esforços para a criação da UML tiveram início em outubro de 1994, quando Rumbaugh se juntou a Booch na *Rational*. Com o objetivo de unificar os métodos Booch e OMT, foi lançado, em outubro de 1995, o esboço da versão 0.8 do *Unified Process* - Processo Unificado (como era conhecido). Naquela mesma época, Jacobson se associou à *Rational*, e o escopo do projeto da UML foi expandido para incorporar o método Object Oriented Software Engineering (OOSE). Nasceu então, em junho de 1996, a “versão 0.9 da UML”.

Atualmente, a UML encontra-se na versão 2.4 e possui 14 diagramas, a saber: diagrama de atividades, diagrama de pacotes, diagrama de sequência, diagrama de comunicação, diagrama de estados, diagrama de componentes, diagrama de implantação, diagrama de tempo ou diagrama de sincronismo, diagrama de classes, diagrama de casos de uso, diagrama de visão geral, diagrama de estrutura composta e diagrama de pacotes.

O fato da UML ser composta por vários diagramas se dá em virtude de fornecer múltiplas visões do sistema a ser modelado, analisando-o e modelando-o sob diversos aspectos, procurando-se, assim, atingir a completude das modelagem, permitindo que cada diagrama complemente os outros [28].

O diagrama de atividades tem o objetivo de especificar o processo de trabalho da empresa, relacionando as atividades executadas no processo aos papéis responsáveis pela sua execução. Ele apresenta um fluxo de dependência de atividades, pois as atividades são sequenciais e mutuamente dependentes para serem executadas.

O diagrama de pacotes é um digrama estrutural, que tem por objetivo representar os subsistemas ou submódulos englobados por um sistema, de forma a determinar as partes que o compõem. Pode ser utilizando também para auxiliar a

demonstrar a arquitetura de uma linguagem, como ocorre com a própria UML, ou ainda para definir as camadas de um software ou processo de desenvolvimento [28].

O diagrama de sequência é utilizado para ilustrar a interação entre os objetos. Eles modelam os objetos e as mensagens entre os objetos [1]. Costumam identificar o evento gerador do processo modelado, bem como o ator responsável por esse evento [28].

O diagrama de comunicação era conhecido como de colaboração até a versão 1.5 da UML, tendo seu nome modificado para diagrama de comunicação, a partir da versão 2.0. Está amplamente associado ao diagrama de sequência. O diagrama de comunicação não se preocupa com a temporalidade do processo, concentrando-se em como os elementos do diagrama estão vinculados e quais mensagens trocam entre si durante o processo [28].

O diagrama de estados demonstra o comportamento por meio de um conjunto finito de transições de estado, ou seja, uma máquina de estado [28].

O diagrama de componentes está associado à linguagem de programação que será utilizada para desenvolver o sistema modelado. Representa os componentes do sistema em termos de módulos de código-fonte, bibliotecas, formulários, entre outros [28].

O diagrama de implantação mostra a configuração dos nós de processamento em tempo de execução e os componentes neles existentes [3].

O diagrama de tempo ou diagrama de sincronismo é um diagrama de interação que mostra os tempos reais em diferentes objetos ou papéis, em vez da sequência de mensagens relativas. É um híbrido do diagrama de atividades e um diagrama de sequência [3].

O diagrama de classe mostra um conjunto de classes, interfaces, colaborações e seus relacionamentos. Este diagrama é importante não só para a visualização, a especificação e a documentação de modelos estruturais, mas também para a construção de sistemas executáveis por intermédio de engenharia de produção e reversa [3].

O diagrama de caso de uso exhibe um conjunto de casos de uso e atores e seus relacionamentos. São importantes principalmente para a organização e a modelagem de comportamento do sistema [3].

O diagrama de visão geral é uma variação do diagrama de atividade que fornece uma visão geral dentro de um sistema ou processo de negócio [28].

O diagrama de estrutura composta descreve a estrutura de um classificador, como uma classe ou componente, detalhando as partes internas que o compõem, como estas se comunicam e colaboram entre si [28].

Dentre os diagramas citados, utilizar-se-ão como objeto de estudo os diagramas de caso de uso, atividades, classe e sequência.

Para melhor entendimento do processo e de como iniciar a modelagem utilizando a UML, são utilizadas as técnicas de *workflow*, juntamente com os mapas conceituais, de modo a facilitar a compreensão para elaboração dos diagramas.

## 2.4 *WORKFLOW*

Esta seção oferece uma introdução à tecnologia de *workflow*, com a finalidade de destacar conceitos importantes tomados como base para o desenvolvimento dos processos propostos desta pesquisa. Serão definidos os conceitos de *workflow*, baseados nos padrões da *Workflow Management Coalition* (WfMC).

A WfMC é uma organização internacional formada em 1993. Seu objetivo é promover a área de *workflow*, por meio da divulgação da tecnologia e do desenvolvimento de padrões para a interoperabilidade de sistemas de *workflow*.

Existem várias definições de *workflow* e *Workflow Management System* (WfMS) na literatura [36] [37] [38], porém com enorme semelhança entre esses conceitos. Com base nas definições da WfMC [34], pode-se definir *workflow* como sendo uma coleção de atividades organizadas para realizar um processo de negócios, por exemplo, para se processar uma nota fiscal ou se tomar uma decisão. Um *workflow* estabelece a ordem de execução das atividades e as condições em que cada atividade pode ser iniciada, assim como a sincronização das atividades e o fluxo de informações.

Os WfMS são sistemas que permitem a definição, a criação e o gerenciamento da execução de *workflows*. Esses sistemas executam uma ou mais máquinas de *workflow*, que são aptas a interpretar a definição dos processos, interagir com os participantes do *workflow* e, quando necessário, invocar ferramentas ou aplicações.

A tecnologia de *workflow* envolve muitos outros conceitos inter-relacionados, conforme demonstra a Figura 2.2.

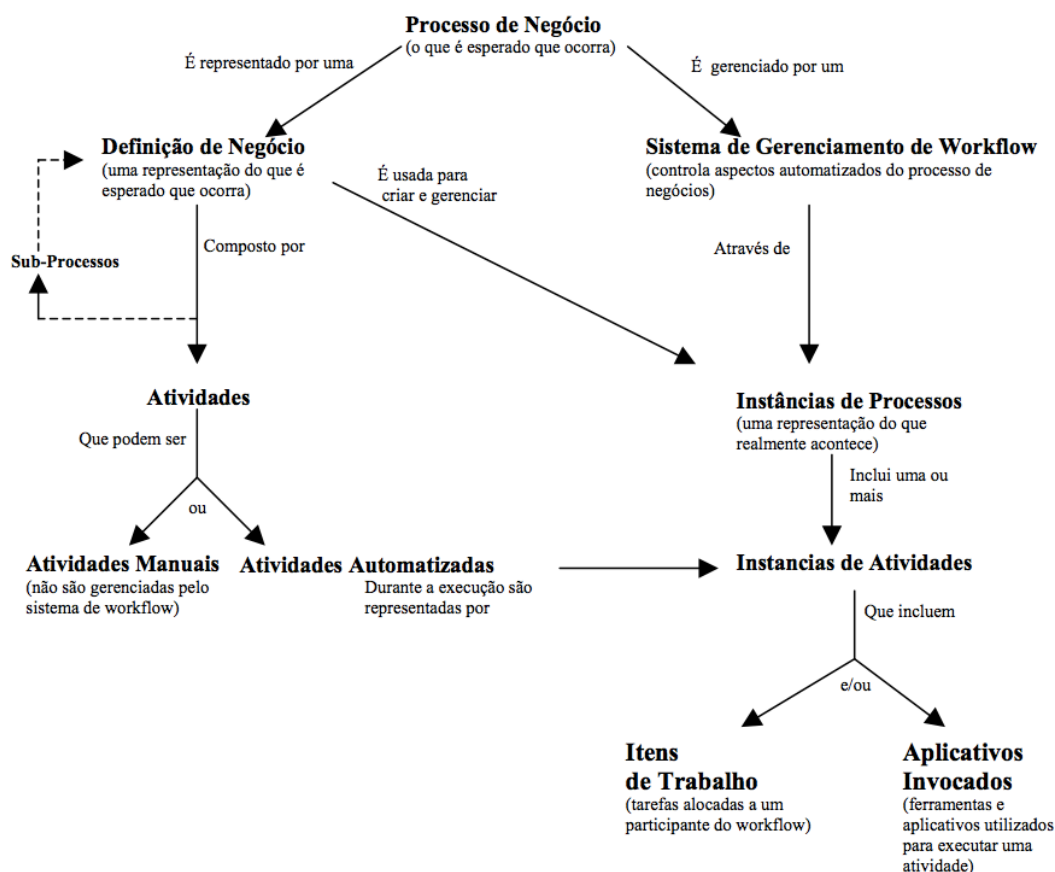


Figura 2.2: Relação da terminologia básica associada com *workflow* [34]

A seguir, são descritos os conceitos básicos relacionados à terminologia utilizada na Figura 2.2, associados à tecnologia de *workflow*.

- **Processo de Negócios**

Processo de negócios, ou simplesmente processo, é um conjunto de um ou mais procedimentos ou atividades relacionadas, que coletivamente atinge um objetivo de negócios, normalmente dentro do contexto de uma estrutura organizacional, que define cargos/funções e relações.

Os processos podem variar significativamente em aspectos como tempo de duração (desde meses até minutos) e abrangência dentro das organizações (desde um único setor até vários setores de diferentes organizações). As atividades englobadas nos processos podem variar de programas de computador até atividades humanas, como, por exemplo, reuniões e tomadas de decisão. Um processo pode ser representado através de uma definição de processo.

- **Definição de Processo**

A definição de um processo é a representação de um processo, de maneira que este suporte manipulações automatizadas, como modelagem ou execução por um sistema de gerenciamento de *workflow*. A definição do processo consiste de uma rede de atividades, seus relacionamentos, critérios para indicar o início e o término dos processos e informações sobre as atividades em si, como por exemplo, participantes, aplicativos utilizados e dados de sistemas de informações relacionados.

- **Subprocesso**

Um processo que é executado ou chamado de outro processo (ou subprocesso), e que faz parte do processo global. Múltiplos níveis de subprocessos podem ser definidos. Um subprocesso é útil para definir componentes reutilizáveis dentro de outros processos. Cada subprocesso tem sua própria definição de processo.

- **Atividade**

Atividade é a descrição de um conjunto de itens de trabalho que realiza um passo lógico dentro de um processo. Uma atividade pode ser manual, que não suporta automação computacional, ou uma atividade automatizada de *workflow*. Uma atividade de *workflow* requer recursos humanos e/ou computacionais para ser executada.

- **Instância de Processo**

É a representação de uma única execução de um processo. Uma instância de processo é criada, gerenciada, e (eventualmente) terminada por um sistema gerenciador de *workflow*, conforme a definição de processo. Cada instância de processo exibe um estado interno que representa a evolução do processo e de suas atividades.

- **Instância de Atividade**

É a representação de uma atividade em uma (única) execução de um processo. Uma instância de atividade é criada e gerenciada por um sistema gerenciador de *workflow*, conforme a definição de processo.

- **Item de Trabalho**

Do ponto de vista do participante do *workflow*, as atividades são representadas como uma coleção de itens de trabalho. Cada item de trabalho (tarefa)

representa uma partição de uma atividade, como a revisão de um documento ou o preenchimento de um formulário.

- **Chamada de Aplicação**

É uma aplicação de *workflow*, que é invocada pelo sistema gerenciador de *workflow* para automatizar, completamente ou parcialmente, uma atividade, ou ajudar um participante de *workflow* no processamento de um item de trabalho. A aplicação pode ser chamada diretamente pelo sistema gerenciador de *workflow* ou pode ser chamada indiretamente por um ator da aplicação.

- **Ator**

Um ator é responsável pela execução parcial ou total de uma determinada instância de atividade. Um ator pode tanto ser humano (por exemplo, um funcionário), quanto um software (por exemplo, um software para envio de fax).

Segundo Casati [40], na execução de um *workflow* têm-se os seguintes tipos de atores:

- **executor do *workflow***: é o ator que inicia (dispara) o *workflow*. Em geral, o *workflow* possui uma lista de possíveis executores;
- **responsável**: é o ator que tem responsabilidade sobre o *workflow*. Pode ser tanto o executor quanto qualquer outra pessoa pertencente à organização;
- **executor da atividade**: é o responsável pela execução da atividade.
- **cargo/função**: cargo/função é um conjunto de atributos associados

aos participantes de um *workflow*. Por exemplo, um processo de desenvolvimento de software poderia ter cargos como analista de sistemas e programador.

#### 2.4.1 Modelos de *Workflow*

As diferenças encontradas na modelagem da maioria dos sistemas comerciais permitem dividir os modelos de *workflow* em dois grupos [36]:

- **modelo baseado em comunicação**: neste modelo, o trabalho é visto como um conjunto de interações humanas bem definidas,



representando compromissos realizados entre as pessoas envolvidas.

- **modelo baseado em atividades:** neste modelo o trabalho é visto como uma sequência de atividades, onde cada atividade recebe um certo conjunto de entradas e produz um certo conjunto de saídas.

#### Modelos de Casati/Ceri

O modelo proposto por Casati, Ceri, Pernici e Pozzi [38] é um dos mais completos para definição de processos. O modelo acrescenta a possibilidade de modelar o acesso a bases de dados externas, via comandos SQL. Assim, os autores visam a integrar sistemas de *workflow* aos sistemas de informação atuais, que fortemente utilizam banco de dados.

Nicolao [38] ressalta que a utilização do modelo de Casati/Ceri está presente em muitos dos trabalhos relacionados à modelagem conceitual de *workflow*. De fato, registra-se considerável número de citações deste modelo em trabalhos, que visam à modelagem conceitual de *workflow*.

No modelo em referência, um esquema de *workflow* consiste em um conjunto de tarefas e um conjunto de conexões entre elas. O modelo descreve quais tarefas podem ser executadas, em qual ordem, quem pode estar encarregado dela, e que operações podem ser executadas. Tarefas podem ser tarefas de trabalho (unidades de trabalho simples), tarefas de roteamento (roteadores de fluxo - tarefas de roteamento são expressas por mecanismos de fluxo de controle análogos aos das linguagens de programação comuns, com a adição de construções que permitam paralelismo), supertarefas (unidades de trabalho complexas, que podem ser decompostas em tarefas de trabalho ou outras supertarefas), ou multitarefas (execuções paralelas de mais cópias da mesma tarefa de trabalho) [38].

#### Modelo de Gatilhos

A técnica de modelagem de *workflow* por Gatilhos foi proposta por Joosten [40]. O uso de gatilhos em sistemas de *workflow* é extremamente relevante, uma

vez que os gatilhos inter-relacionam as atividades, que são executadas através de máquinas e/ou pessoas.

Um *workflow* no Modelo de Gatilhos é definido em termos dos seguintes elementos básicos:

- **evento:** algo que acontece; algo que ocorre (Ex.: a chegada de uma carta que é postada);
- **ator:** alguém que atua (Ex.: a pessoa que posta uma carta);
- **objeto:** algo que é capaz de ser visto, tocado ou sentido (Ex.: a carta);
- **atividade:** conjunto de eventos que ocorrem sob a responsabilidade de um ator. Atividades são executadas como resultado da ocorrência de eventos;
- **gatilho:** um evento **e** é um gatilho para uma atividade **a**, se a ocorrência de **e** ocasiona a execução de **a**. Na oração “**a** dispara **b**”, **a** pode ser um evento, uma atividade ou um ator, mas **b** é sempre uma atividade;
- **processo:** conjunto de atividades que compartilham um propósito comum. Processos são definidos para dar nome a um conjunto de atividades que estão relacionadas e que fazem sentido em uma situação específica. A diferença entre atividade e processo está no fato de o processo poder envolver diferentes atores responsáveis, enquanto a atividade possui um ator que tem a responsabilidade pela execução;
- **workflow:** sistema cujos elementos são atividades, inter-relacionadas por gatilhos e disparadas através de eventos externos (Ex.: trabalho causado em um hospital, quando um paciente novo é admitido);
- **cargo/função:** conjunto de atores que desempenham as mesmas funções (Ex.: clientes, gerentes, atendentes).

## Modelo de Ações

No modelo de ações, todas as ações realizadas no *workflow* são reduzidas a um conjunto limitado de atos da fala, classificados e ordenados de determinada forma e que representam as interações possíveis entre o cliente e o fornecedor do processo.

A partir da teoria dos atos da fala e da compreensão dos autores sobre como se desenrola o processo de comunicação dentro das organizações, foi proposto um *loop* básico de trabalho, também denominado laço de *workflow*. Esse laço representa fielmente a estrutura da comunicação humana, sendo genérico para qualquer situação de trabalho que se deseje modelar e universal, no sentido em que é independente de qualquer cultura, linguagem ou meio de comunicação utilizado para conduzi-lo [39]. Esse laço identifica quatro fases na comunicação entre o cliente (aquele que solicita que algo seja realizado) e o fornecedor (aquele que executará algo para o cliente) do processo.

### 2.5 INTEGRAÇÃO WORKFLOW E MAPAS CONCEITUAIS

Para facilitar o processo de ensino e aprendizagem da UML, foi proposta a utilização do *workflow* e do mapa conceitual. O *workflow* demonstra a dinamicidade do processo de elaboração dos diagramas. Porém, o processo de aprendizagem torna-se completo somente com o uso dos mapas conceituais, uma vez que a união das duas tecnologias faz com que o processo seja completo e significativo. Tem-se que “a ideia básica de um *workflow* é a quebra do processo em partes menores, sendo estas denominadas de atividades” [32].

Objetivando a utilização da aprendizagem significativa, foi desenvolvido um mapa conceitual para cada diagrama da UML e também foram modelados um ou vários processos de *workflows*, onde demonstra-se as atividades a serem seguidas para a elaboração dos mesmos.

Sizilio [8] afirma que a tecnologia de *workflow* tem-se apresentado como possibilidade de modelar as atividades inerentes ao ensino, em virtude da clareza de definição das tarefas a serem executadas com seus agentes responsáveis.

O *workflow* foi utilizado para tornar dinâmico o mapa conceitual, isto é, por meio do *workflow* foi demonstrado o fluxo de atividades para o desenvolvimento de um dado diagrama, no qual as informações tramitarão entre os atores envolvidos, gerando

artefatos. Cada atividade do *workflow* deve prover produtos advindos das atividades anteriores, que serão utilizados na atividade corrente, bem como a metodologia a ser utilizada nesta atividade. Nesta, por meio de uma instrução de trabalho, são descritos os recursos necessários (Recursos Humanos, Máquinas, Software, entre outros) e os produtos resultantes (Artefatos), conforme demonstrado na Figura 2.3.

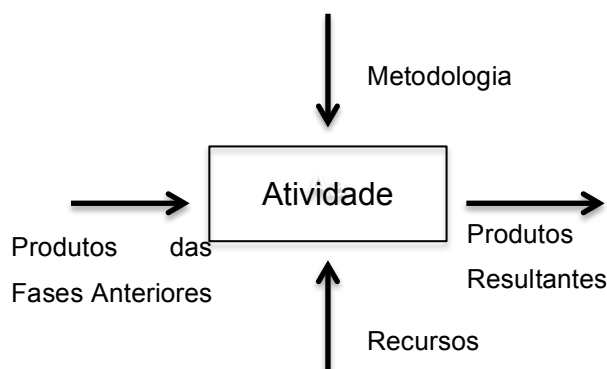


Figura 2.3: Elementos que compõem as atividades

Com essas informações em mãos, tem-se um mecanismo de auxílio ao processo de ensino-aprendizagem, que permite tanto ao professor quanto ao aluno se posicionarem e compreenderem de maneira mais sistêmica os diagramas da UML.

## 2.6 TRABALHOS RELACIONADOS

A utilização das técnicas de *workflow* foi encontrada em alguns trabalhos, como também foram localizados trabalhos relacionados com a UML. Em Lopes [14], foi abordada a tecnologia de *workflow* vinculada à aprendizagem, trazendo uma proposta de integração de técnicas de Planejamento em Inteligência Artificial (IA) e tecnologia de *workflow* a um ambiente de Ensino a Distância, onde a tecnologia de *workflow* foi apresentada e técnicas de planejamento de IA, utilizadas. Lopes percebeu a necessidade de monitorar os dados e direcionar as atividades de cada estudante, visando assim a auxiliar o desempenho do aluno no curso e no aprendizado. Lopes ainda propôs um sistema de gerenciamento do processo de aprendizagem, baseado em um *workflow*, capaz de atuar como um gerente automático para auxiliar no planejamento e execução dos conteúdos e no monitoramento do progresso do estudante do curso em EAD.

Pichiliani [12], por sua vez, mostra como utilizar a modelagem colaborativa no aprendizado a UML. Sua proposta é uma alternativa para professores avaliarem, acompanharem a aprendizagem e dificuldades de grupos de alunos que utilizaram uma ferramenta colaborativa para auxiliar o aprendizado da UML.

Foram localizados trabalhos sobre mapas conceituais sendo utilizados no processo de ensino aprendizagem [3], bem como trabalhos sobre a UML e também a utilização do *workflow* [6], [8], [15]. Entretanto, nenhum deles aborda as técnicas agrupadas afim de contribuir para o processo de ensino e aprendizagem da UML.

## 2.7 CONSIDERAÇÕES FINAIS

A aprendizagem sempre será um tema de grande relevância. O professor preocupa-se com “o quanto” o aluno está assimilando o conteúdo ministrado, enquanto o aluno necessita de forma que facilite esse processo.

A UML é uma linguagem de modelagem mundialmente utilizada. Portanto, torna-se crescente a necessidade de ministrar esse conteúdo de forma mais simples e clara possível, a fim de facilitar o processo de aprendizagem do aluno.

Encontrou-se na tecnologia de *workflow* e de mapas conceituais maneira de facilitar o processo de ensino e aprendizagem da UML. Esta aprendizagem deve despertar no aluno a vontade de aprender, de forma significativa e não utilizando a técnica de memorização.

Foram localizados trabalhos relacionados à UML e seus diagramas e também muitos trabalhos em relação ao *workflow* (principalmente relacionado a processo de negócio). Porém, poucos foram aqueles que relacionam o *workflow* com a aprendizagem. Entretanto, não foram localizados trabalhos abordando as técnicas em conjunto.

### 3 IMPLEMENTAÇÃO DO MAPA CONCEITUAL E DO *WORKFLOW*

De acordo com a análise da dificuldade dos alunos em assimilar a UML, bem como a do professor em ministrar o conteúdo, foram elaborados mapas conceituais com o objetivo de facilitar o processo de ensino e aprendizagem.

Visando a auxiliar na maneira de ensinar e aprender os diagramas da UML, são apresentados um modelo de mapa conceitual para demonstrar a relação entre os diagramas e outro para tratar de forma mais detalhada os conceitos referentes a cada diagrama que faz parte do objeto de estudo.

Nicolao [38] concluiu que o modelo de Casati teve significativa contribuição, abordando a noção de modularização (supertarefas), além de construções básicas de paralelismo (também presentes no modelo de gatilhos), tais como *Fork e Join*.

Para a elaboração do *workflow*, foi utilizada a ferramenta *BizAgi Process Modeler* [41], cujas notações são semelhantes à notação adotada por Casati.

#### 3.1 ESTRUTURA DOS DIAGRAMAS DA UML

Para orientação quanto à elaboração dos diagramas da UML, em relação a “o que fazer primeiro”, foi elaborada uma estrutura que demonstra os diagramas da UML existentes. Lima [17] afirma que, “dependendo do porte do sistema, nem todos os diagramas são desenhados, mas qualquer projeto possui, pelo menos, diagrama de caso de uso, de interação e de classe”.

Booch [3] compara a construção de uma casa com a construção de um software, onde em cada etapa de uma casa, o arquiteto necessita visualizar o projeto com perspectivas diferentes, a fim de esclarecer detalhes. Similarmente, os diagramas são criados incrementalmente (ampliando-os, uma parte de cada vez) e iterativamente (repetindo o processo de projetar uma pequena parte e construí-la).

Como se observa na Figura 3.1, pode-se iniciar pela elaboração do diagrama de implantação, do diagrama de pacotes, do diagrama de caso de uso, e do diagrama de atividades, de forma independente dos demais diagramas. Após se iniciar a elaboração do diagrama de caso de uso, onde são modelados os requisitos do

comportamento do sistema [3], é possível, também, dar início ao diagrama de classe e diagrama de atividades.

Durante a elaboração dos diagramas, podem ocorrer refinamentos sucessivos, representados por um retângulo pontilhado sobre os diagramas que sofrem tais refinamentos, como o diagrama de caso de uso, diagrama de classe e diagrama de atividades. O diagrama de classe encontra-se no centro do processo de modelagem de objetos [1]. Com este diagrama definido, outros diagramas podem ser elaborados, pois todos eles possuem dependência em relação às classes.

Para que a aprendizagem seja realmente significativa, somente a estrutura hierárquica dos diagramas da UML apresentada na Figura 3.1 não é suficiente, visto que a mesma está mostrando as relações dos diagramas e não o que é necessário fazer, passo-a-passo, para construção de cada diagrama, ou seja, a dinamicidade para o processo de ensino e aprendizagem. Para completar a forma de aprendizagem, foram desenvolvidos o *workflow* e os mapas conceituais para os diagramas pesquisados.

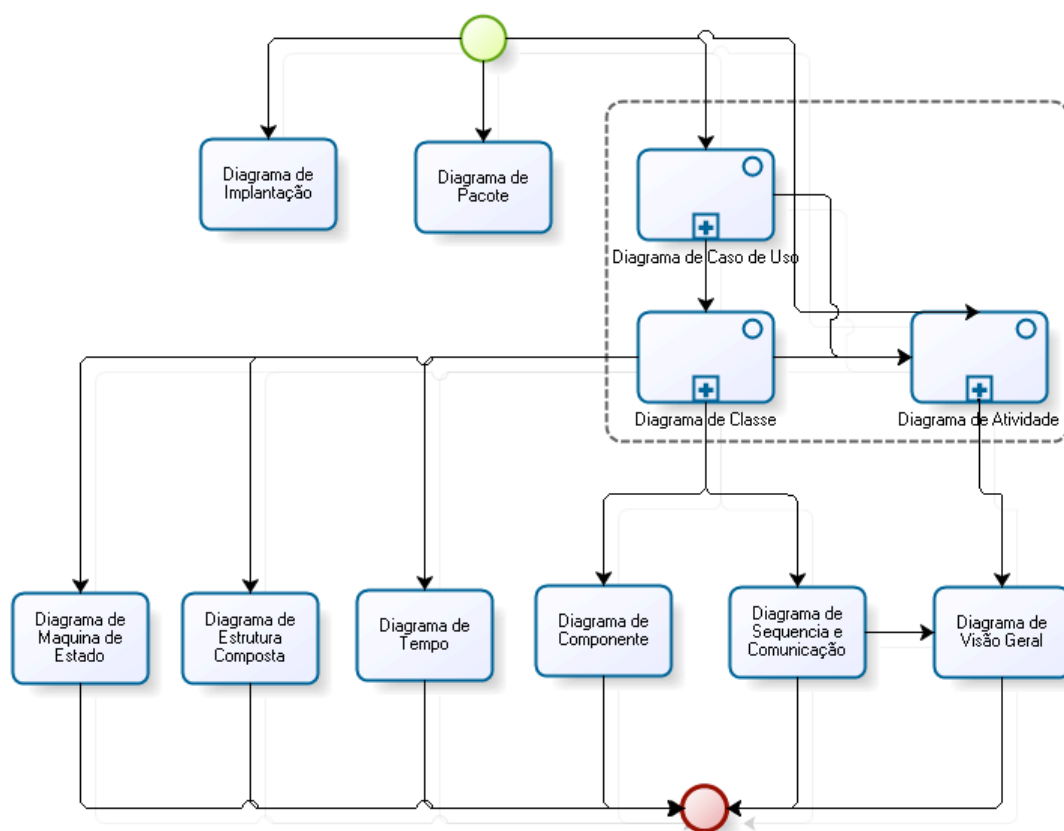


Figura 3.1: Estrutura da Elaboração dos Diagramas da UML.

### 3.2 MAPA CONCEITUAL E *WORKFLOW* DO DIAGRAMA DE CASO DE USO

Um diagrama de caso de uso é um diagrama que mostra um conjunto de casos de uso, atores e seus relacionamentos [3]. Os casos de usos são uma descrição de um conjunto de sequência e ações que um sistema executa para produzir algo de valor observável para o ator [3]. Eles representam os requisitos funcionais do sistema e envolvem a interação do ator com o sistema.

O ator frequentemente representa um papel que um ser humano exerce. Contudo, pode ser um hardware ou até mesmo outro sistema, apesar da sua representação demonstrar um “homem-palito”.



Um Caso de Uso descreve o que um sistema faz, mas não como ele faz [3]. A simplicidade do diagrama de caso de uso produz pontos fortes e fracos. Um ponto forte é a falta de explicação. Para que esse ponto fraco seja resolvido, o caso de uso normalmente é acompanhado por uma descrição narrativa ou também chamado de especificação de caso de uso [1].

Na Figura 3.2 é demonstrado o *workflow* para a elaboração do diagrama de caso de uso e na Figura 3.3 é demonstrado de forma mais detalhada a atividade “Fazer o Diagrama de Caso de Uso”.

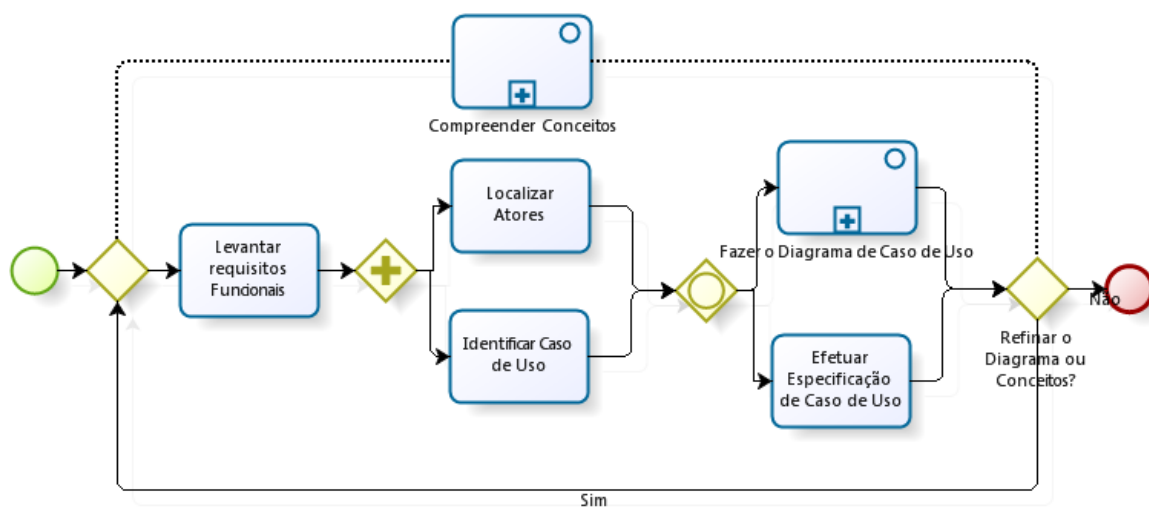


Figura 3.2: *Workflow* para elaboração do Diagrama de Caso de Uso.

Como pode se verificar na Figura 3.2, a elaboração do diagrama de caso de uso inicia-se com uma decisão em relação à compreensão dos conceitos relacionados com o diagrama de caso de uso ou não. Se a decisão for “Sim”, será exibido os mapas conceituais referente ao diagrama de caso de uso, caso contrário iniciará com o levantamento de requisitos.

Posteriormente, há duas atividades que acontecem paralelamente: a localização dos atores e dos casos de uso. Ao término das duas atividades, iniciam-se outras duas atividades paralelas: a modelagem do diagrama de caso de uso, com suas respectivas especificações. Para um bom entendimento do problema e para agregar qualidade ao desenvolvimento do software, é necessária a realização de refinamentos sucessivos. Isto é representado no *workflow* pelo *fork* condicional, que acontece após a execução das duas atividades anteriores, retornando ou não para o início do *workflow*.

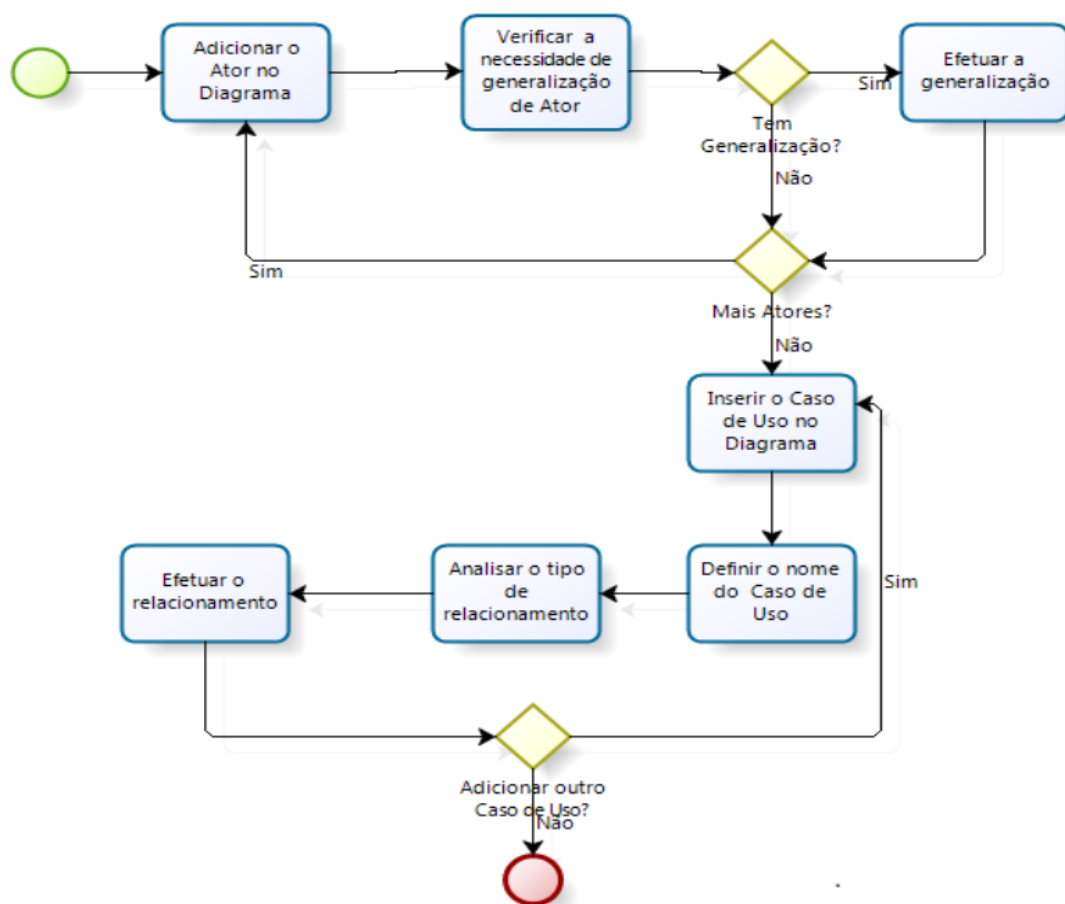


Figura 3.3: *Workflow* da Atividade Fazer Diagrama de Caso de Uso (Diagrama de Caso de Uso).

A Figura 3.3 demonstra o *workflow* da atividade “Fazer o Diagrama de Caso de Uso”, que visa a detalhar passo a passo como elaborar o Diagrama de Caso de Uso, dando sequência às atividades “Levantar requisitos funcionais”, “Localizar Atores” e “Identificar Caso de Uso”, já efetuadas.

Para elaborar o diagrama de caso de uso, deve-se adicionar nele o ator (já identificado). Em seguida, verifica-se a necessidade de generalização do ator. Identificada a necessidade de generalização, esta deve ser feita, caso contrário, verifica-se a necessidade de adicionar novos atores, até que seja encerrada a adição de atores. A propósito de continuidade, deve-se inserir o caso de uso (já identificado), nomeá-lo, analisar o tipo de relacionamento e, finalmente, relacionar os casos de uso e atores. Na sequência, verifica-se se existem outros casos de uso e executa-se o mesmo procedimento, até que não existam mais casos de usos a serem inseridos.

Conforme ilustrado na Figura 2.3, o *workflow* é composto por atividades, e vários itens têm influenciado nestas atividades, com o intuito de resultar em um produto. No *workflow* do diagrama de caso de uso, foram identificadas cinco atividades, sendo que algumas delas podem possuir subatividades. Todas as atividades estão descritas com detalhes nos Quadros de 1 a 5.

<p align="center"><b>Atividade “Levantar Requisitos Funcionais”</b></p> <p>A análise de requisitos é o primeiro passo técnico do processo de Engenharia de Software. É nesse ponto que uma declaração geral do escopo do software é aprimorada numa especificação concreta que se torna a base para todas as atividades de Engenharia de Software que surgirão [18].</p> <p>Um Caso de Uso representa um requisito funcional do sistema [3], portanto o levantamento de requisitos, em especial, os requisitos funcionais, é de extrema importância para elaboração do diagrama de Caso de Uso.</p>
<b>Metodologia</b>
<p>Entrevistas.</p> <p>Questionários.</p> <p>Ações dos atores.</p> <p>Simulação do dia-a-dia do usuário.</p>
<b>Produtos das Fases Anteriores</b>
<p>Conhecimento do Negócio por intermédio do levantamento de requisitos. .</p>
<b>Produtos Resultantes</b>
<p>Documentos (como exemplo):</p> <p>“Documento de Visão”, que traz uma visão geral do sistema a ser desenvolvido.</p> <p>“Solicitação dos principais envolvidos” em que consta o resultado das entrevistas, entre outras informações.</p> <p>Especificação complementar, que detalha os requisitos não-funcionais.</p>
<b>Recursos</b>
<p>Analista de Sistemas.</p> <p>Usuário.</p> <p>Computador.</p> <p>Software Editor de Texto.</p>

Quadro 1: Atividade “Levantar Requisitos Funcionais”

Atividade “Localizar Atores”	
Um ator especifica um papel desempenhado por um usuário ou qualquer outro sistema que interage com o sujeito, isto é, qualquer entidade externa que interage com o sistema a ser modelado [17].	
Conforme RUP [19], a localização de atores é um dos primeiros passos da definição do uso do sistema. Cada tipo de fenômeno externo com o qual o sistema deverá interagir é representado por um ator. Para localizar os atores, faça as seguintes perguntas:	
<ul style="list-style-type: none"> <li>• Quais grupos de usuários necessitam de ajuda do sistema para executar tarefas?</li> <li>• Quais grupos de usuários são necessários para executar as funções básicas mais óbvias do sistema?</li> <li>• Quais grupos de usuários deverão executar funções secundárias, como manutenção e administração do sistema?</li> <li>• O sistema interagirá com algum sistema externo de <i>hardware</i> ou <i>software</i>?</li> </ul>	
Para determinar se possuem os atores (humanos) adequados, você poderá tentar nomear duas ou três pessoas que podem atuar como atores e ver se o seu conjunto de atores é suficiente para as necessidades deles.	
Metodologia	
Simulação do dia-a-dia do usuário.	
Produtos das Fases Anteriores	
Conhecimento do Negócio por intermédio do levantamento de requisitos. Solicitação dos principais envolvidos. Documento de Visão.	
Produtos Resultantes	
Identificação dos Atores.	
Recursos	
Analista de Sistemas. Usuário. Computador. Software Editor de Texto.	

Quadro 2: Atividade “Localizar Atores”.

<p style="text-align: center;">Atividade “Identificar os Casos de Uso”</p> <p>Os Casos de Uso, com certeza, sofre alteração. Novos Casos de Uso podem surgir, bem como os que já existem podem ser alterados ou eliminados.</p> <p>A definição do Caso de Uso depende do levantamento de requisitos, uma vez que, se bem realizado, facilitará a identificação dos Casos de Uso.</p> <p>Ao se efetuar o levantamento de requisitos e das solicitações dos envolvidos no projeto, é possível identificar os atores, sendo assim, identificar o relacionamento dos atores com o sistema, facilitando a identificação do Caso de Uso.</p> <p>De acordo com RUP 7.0 [20], alguns questionamentos são primordiais para identificar os Casos de Uso e as repostas para estas perguntas são sugestões de Caso de Uso. Para isso, cada ator, humano ou não, deve fazer a si mesmo estas perguntas:</p> <ul style="list-style-type: none"> <li>• Segundo o ator, quais são as principais tarefas a serem executadas pelo sistema?</li> <li>• O ator criará, armazenará, alterará, removerá ou lerá dados no sistema?</li> <li>• O ator precisa informar o sistema sobre mudanças externas repentinas?</li> <li>• O ator precisa estar informado sobre certas ocorrências no sistema?</li> <li>• O ator inicializará ou desligará o sistema?</li> </ul>
Metodologia
<p>Entrevistas.</p> <p>Questionários.</p> <p>Ações dos atores.</p> <p>Simulação do dia-a-dia do usuário.</p>
Produtos das Fases Anteriores
<p>Conhecimento do Negócio por intermédio do levantamento de requisitos. .</p> <p>Documentos (como exemplo, o “Documento de Visão” que traz uma visão geral do sistema a ser desenvolvido).</p> <p>“Solicitação dos principais envolvidos”, onde consta o resultado das entrevistas, entre outras informações.</p>
Produtos Resultantes

<ul style="list-style-type: none"> <li>• Diagrama de (visão preliminar)</li> <li>• Especificação de Caso de Uso (visão preliminar)</li> </ul> <p>(Obs.: O Diagrama de Caso de Uso e a Especificação de Caso de Uso são citados como visão preliminar, pois os mesmos sofrem recursividade, conforme demonstrados na Figura 3.2. Sendo assim, serão refinados, tornando-se completos.</p>
Recursos
<p>Analista de Sistemas.</p> <p>Usuário.</p> <p>Computador.</p> <p>Software Editor de Texto.</p> <p>Software de diagramação (ex. Rational Rose, Jude, ArgoUML).</p>

Quadro 3: Atividade “Identificar Caso de Uso”.

Os diagramas com atores, casos de uso e relacionamentos entre eles são denominados Diagramas de Casos de Uso e ilustram os relacionamentos no modelo de Casos de Uso. Os diagramas de casos de uso podem ser organizados em (e pertencentes a) pacotes de Casos de Uso, mostrando apenas o que é relevante em determinado pacote [20].

Um Diagrama de Caso de Uso bem estruturado [3]:

- Tem como foco comunicar um aspecto da visão estática de Caso de Uso do sistema;
- contém somente os Casos de Uso e atores essenciais à compreensão desse aspecto.

Ao efetuar os diagramas de Caso de Uso [3]:

- Dê um nome capaz de comunicar seu propósito;
- evite o cruzamento de linhas;
- organize os elementos espacialmente.

Após efetuado o levantamento dos atores e Casos de Uso, é possível elaborar o Diagrama de Caso de Uso. Para isso, é necessário inserir o ator no diagrama e verificar a necessidade de generalização entre atores.

Analizados os atores, inseridos no diagrama os Casos de Uso, em seguida, deve-se efetuar o relacionamento entre atores e Caso de Uso e, por fim, refinar o diagrama, analisando a necessidade de Casos de Uso de inclusão e/ou alteração.

Metodologia
Entrevistas. Questionários. Ações dos atores. Simulação do dia-a-dia do usuário.
Produtos das Fases Anteriores
Levantamento de requisitos. Localização de atores. Identificação de Caso de Uso.
Produtos Resultantes
<ul style="list-style-type: none"> <li>• Diagrama de Caso de Uso (visão preliminar).</li> <li>• Especificação de Caso de Uso (visão preliminar).</li> </ul> <p>(Obs.: O Diagrama de Caso de Uso e a Especificação de Caso de Uso são citados como visão preliminar, pois sofrem recursividade, conforme demonstrado na Figura 3.2, sendo assim serão refinados e tornando-se completos.)</p>
Recursos
Analista de Sistemas. Usuário. Computador. Software Editor de Texto. Software de diagramação (ex. Rational Rose, Jude, ArgoUML).

Quadro 4: Fazer o Diagrama de Caso de Uso.

Um Caso de Uso descreve o que o sistema faz, mas não especifica como isso é feito. Você pode especificar o comportamento de um Caso de Uso pela descrição do fluxo de eventos no texto de maneira suficientemente clara para que alguém de fora possa compreendê-lo facilmente [3].

Ao escrever um fluxo de eventos, você deverá incluir como e quando o Caso de Uso inicia ou termina, quando o Caso de Uso interage com os atores e quais objetos são transferidos [3].

A especificação de Caso de Uso tem a finalidade de detalhá-lo, descrevendo os fluxos básicos e alternativos de um determinado Caso de Uso. Através dela, podemos ainda, visualizar seu cenário e uma breve descrição de sua funcionalidade em relação ao sistema. Nela, podemos

encontrar ainda um protótipo do formulário gerado como resultado do Caso de Uso.
<b>Metodologia</b>
Ações dos atores. Simulação do dia-a-dia do usuário.
<b>Produtos das Fases Anteriores</b>
Levantamento de requisitos. Localização de atores. Identificação de Caso de Uso. Diagrama de Caso de Uso.
<b>Produtos Resultantes</b>
<ul style="list-style-type: none"> <li>Especificação de Caso de Uso (visão preliminar).</li> </ul> (Obs.: Especificação de Caso de Uso são citados como visão preliminar, pois os mesmos sofrem uma recursividade, conforme demonstrados na Figura 3.2, sendo assim serão refinados, tornando-se completos.
<b>Recursos</b>
Analista de Sistemas. Usuário. Computador. Software Editor de Texto. Software de Controle de Versão.

Quadro 5: Efetuar a Especificação de Caso de Uso.

Na Figura 3.4, pode-se observar o mapa conceitual do diagrama de caso de uso, o qual apresenta os conceitos envolvidos para o desenvolvimento deste. Para todos os conceitos do mapa conceitual, foi elaborado uma tabela com a definição e notação conforme o Apêndice B.



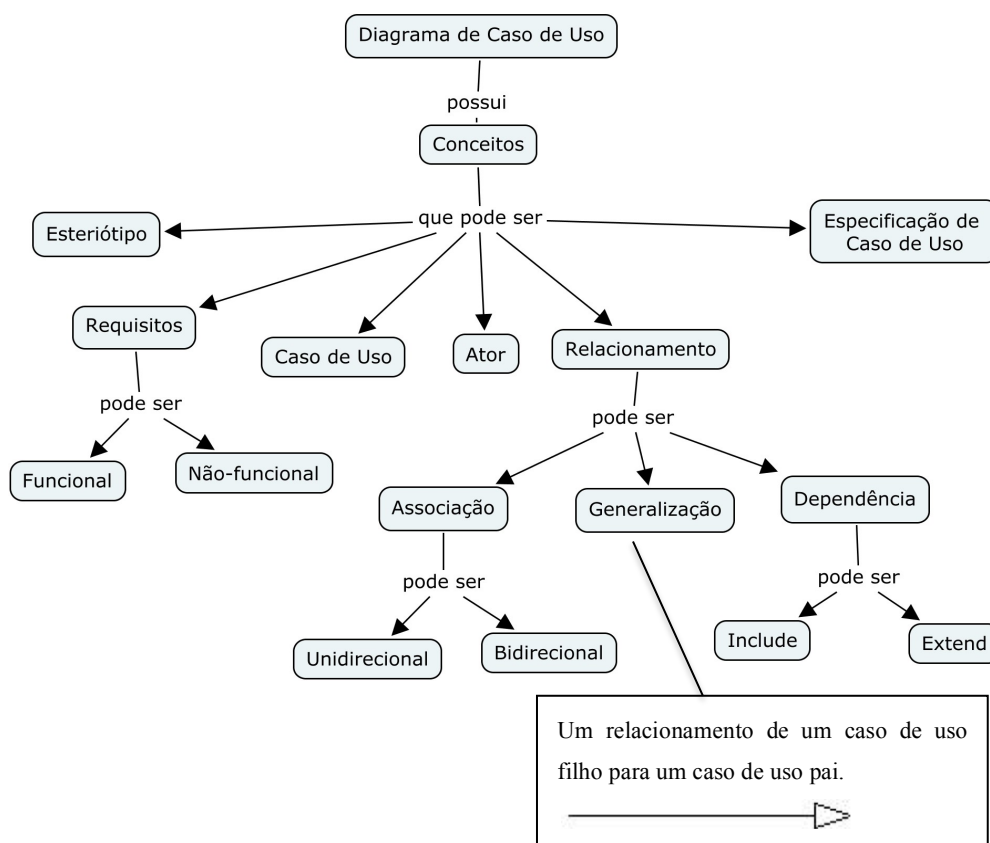


Figura 3.4: Mapa Conceitual do Diagrama de Caso de Uso.

### 3.3 MAPA CONCEITUAL E *WORKFLOW* DO DIAGRAMA DE ATIVIDADES

O diagrama de atividades normalmente é visto como parte da visão funcional de um sistema, pois descreve processos lógicos ou funções. Cada processo descreve uma sequência de tarefas e decisões que controlam quando e como elas são realizadas [1].

As atividades efetivamente resultam em alguma ação, que culminam em mudança de estado do sistema ou o retorno de um valor. As ações abrangem a chamada a outras operações, enviando um sinal, criando ou destruindo um objeto ou alguma computação pura, como cálculo de uma expressão [3].

O diagrama de atividades pode ser empregado a um sistema como todo, a uma operação, a uma classe e também ao caso de uso (para modelagem de um cenário).

A fim de demonstrar como elaborar o diagrama de atividades, foram desenvolvidos o *workflow* e o mapa conceitual, para agregar valor ao cognitivo dos alunos, ou seja, conceitos sem os quais o aluno pode ter maior dificuldade para elaborar tais diagramas.

Como se vê na Figura 3.5, a elaboração do diagrama de atividades inicia-se com uma decisão em relação à compreensão dos conceitos relacionados com o diagrama de atividades ou não. Se a decisão for “Sim”, será exibido os mapas conceituais referente ao diagrama de atividades, caso contrário iniciará pela atividade “Estabelecer o Foco do Diagrama”. Posteriormente, têm-se duas atividades que acontecem paralelamente: “Identificar Grupos e/ou responsáveis” e “Identificar as Atividades”. Posteriormente, inicia-se a atividade “Elaborar o Diagrama de Atividades”, considerada uma supertarefa, demonstrada detalhadamente na Figura 3.6. Para entendimento do problema e para agregar qualidade ao desenvolvimento do software, é necessária a realização de refinamentos sucessivos. Isto é representado no *workflow* pelo *fork* condicional, que acontece após a execução da supertarefa, retornando ou não para o início do *workflow*.

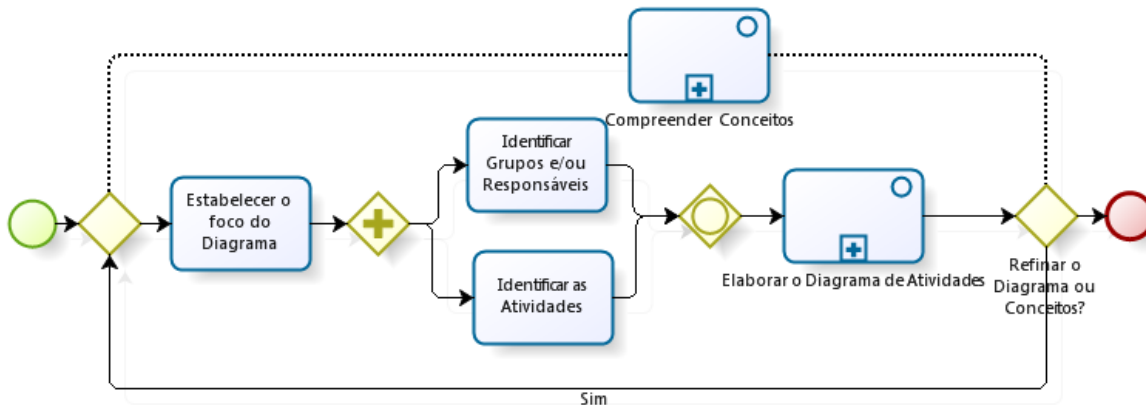


Figura 3.5: *Workflow* do Diagrama de Atividades.

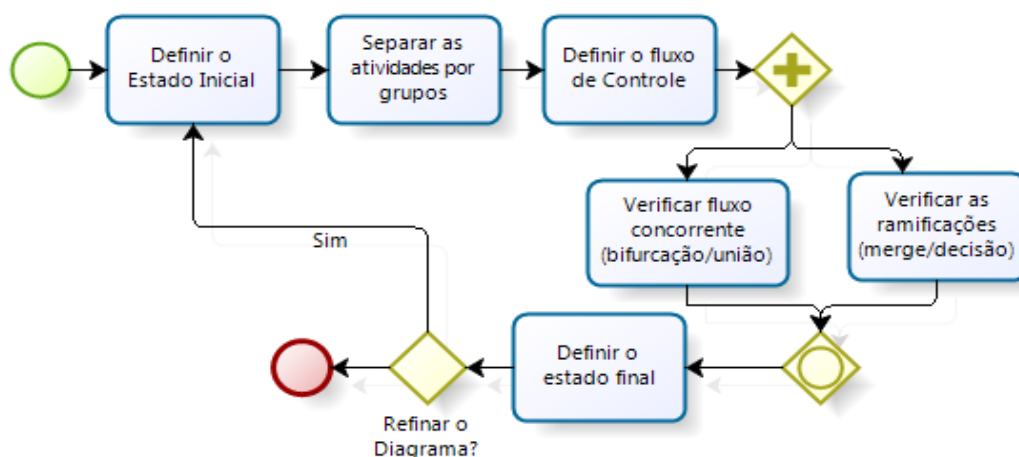


Figura 3.6: *Workflow* da Supertarefa Atividade Elaborar o Diagrama de Atividades.

Existem pelo menos três lugares em um modelo onde um diagrama de atividades oferece ideias valiosas, a saber: “modelando um *workflow*, descrevendo um caso de uso, especificando as operações [1]. Dessa forma, na atividade em questão deve-se definir para qual “lugar” o diagrama de atividades será modelado, antes que outras atividades do *workflow* sejam executadas.

Quando definido pela modelagem do diagrama de atividades para um caso de uso, deve-se levar em consideração que se pode modelar um único caso de uso ou apenas uma parte dele ou até mesmo vários Casos de Uso unidos para criar um *workflow*. Para esta opção, fica claro como o ator interage com o sistema para realizar o objetivo do caso de uso [1]. Ao definir pela elaboração do diagrama de atividades para um determinado método, modelando a sequência de acontecimento, lógicas de decisão, *loops* e, finalmente, ao se elaborar o diagrama de atividades no nível de *workflow*, são empregados no contexto do sistema como um todo [3].

Conforme ilustrado nas Figuras 3.5 e 3.6, o *workflow* é composto por atividade, e vários itens influenciam nas atividades com intuito de resultar em um produto. No *workflow* do diagrama de atividades, foram identificadas quatro atividades, sendo que algumas delas podem possuir subatividades. Seguem os Quadros onde se detalham as atividades, bem como os elementos que compõem as mesmas, conforme demonstrado na Figura 2.3.

Atividade “Estabelecer o Foco do Diagrama”	
<p>Conforme Pender [1], há pelo menos três lugares em um modelo onde um diagrama de Atividades oferece ideias valiosas, sendo eles, “modelando um <i>workflow</i>, descrevendo um Caso de Uso, especificando as operações.”</p> <p>Dessa forma, na atividade em questão deve-se definir para qual “lugar” o diagrama de atividades será modelado, antes que outras atividades do <i>workflow</i> sejam executadas.</p> <p>Quando definido pela modelagem do diagrama de atividades para um Caso de Uso, deve-se levar em consideração que se pode modelar um único Caso de Uso ou apenas uma parte dele ou até mesmo vários Casos de Uso unidos para criar um <i>workflow</i>. Para esta opção, fica claro como o ator interage com o sistema para realizar o objetivo do Caso de Uso [1].</p> <p>Ao definir pela elaboração do diagrama de atividades para um determinado método, modela-se a sequência de acontecimento, lógicas de decisão e <i>loops</i>.</p> <p>Ao elaborar o diagrama de atividades no nível de <i>workflow</i> são empregados no contexto do sistema como um todo [3].</p>	
Metodologia	
<p>Ações dos atores.</p> <p>Simulação do dia-a-dia do usuário.</p>	
Produtos das Fases Anteriores	
<p>Conhecimento do Negócio por intermédio do levantamento de requisitos.</p> <p>Documentos (como exemplo, o “Documento de Visão”, que traz visão geral do sistema a ser desenvolvido, “Solicitação dos principais envolvidos” em que consta o resultado das entrevistas, entre outras informações.)</p> <p>Conhecimento do método, caso o diagrama seja para um método.</p>	
Produtos Resultantes	
Diagrama de Atividades (visão preliminar).	
Recursos	
<p>Analista de Sistemas.</p> <p>Usuário.</p> <p>Computador.</p> <p>Software de diagramação (ex. Rational Rose, Jude, ArgoUML).</p>	

Quadro 6: Atividade “Estabelecer o Foco do Diagrama”.

Ao efetuar o diagrama de atividades, as atividades poderão ficar independentes de quem as realiza. Porém, o ideal é que as tarefas sejam atribuídas a um responsável ou a um grupo de responsáveis.

Na UML, cada grupo ou responsável é chamado de raia de natação, pois, visualmente, os grupos ficam separados por uma linha vertical, com o nome do responsável na parte superior. A raia de natação especifica um conjunto de atividades que compartilham algumas propriedades organizacionais [3].

Cada raia possui um único nome e representa uma responsabilidade. Cada atividade pertence a apenas uma raia.

#### Metodologia

Conhecimento do negócio.

#### Produtos das Fases Anteriores

Diagrama de Caso de Uso.

Especificação de Caso de Uso.

#### Produtos Resultantes

- Diagrama de Atividades (visão preliminar).

#### Recursos

Analista de Sistemas.

Usuário.

Computador.

Software de diagramação (ex. Rational Rose, Jude, ArgoUML).

Quadro 7: Identificar Grupos e/ou responsáveis.

A atividade é uma etapa no processo, onde algum trabalho está sendo realizado. Este trabalho pode ser um cálculo, a localização de algum dado, entre outros [1].

Uma atividade indica uma ação a ser executada.

#### Metodologia

Conhecimento do negócio.

Produtos das Fases Anteriores
Diagrama de Caso de Uso Especificação de Caso de Uso Documento de visão. Levantamento de Requisitos.
Produtos Resultantes
<ul style="list-style-type: none"> <li>• Diagrama de Atividades (visão preliminar).</li> </ul>
Recursos
Analista de Sistemas. Usuário. Computador. Software de diagramação (ex. Rational Rose, Jude, ArgoUML).

Quadro 8: Identificar as Atividades.

<p>Os diagramas de atividade podem ser anexados a qualquer elemento da modelagem com o propósito de visualizar, especificar, construir e documentar o comportamento desse elemento [3].</p> <p>Ao elaborar o diagrama de atividades, conforme Booch [3], deve-se:</p> <ul style="list-style-type: none"> <li>• estabelecer o foco para o fluxo de trabalho. É impossível mostrar todos os fluxos de trabalho em um mesmo diagrama;</li> <li>• criar as raias de natação para cada objeto importante;</li> <li>• minimizar os cruzamentos de linhas;</li> <li>• verificar a necessidade de bifurcação e/ou união.</li> </ul> <p>Ao elaborar o diagrama de atividades, inicia-se o diagrama com a notação de início. Em seguida, inserem-se as atividades e o fluxo de controle. Dando sequência, definem-se as raias com as responsabilidades e analisar a necessidade de bifurcação/união. Para finalizar o diagrama, insere-se a notação de fim.</p>
Metodologia
Conhecimento do negócio.
Produtos das Fases Anteriores

<p>Diagrama de Caso de Uso.</p> <p>Especificação de Caso de Uso.</p> <p>Identificação dos grupos e/ou responsáveis.</p> <p>Identificação das atividades.</p> <p>Estabelecimento do Foco do Diagrama.</p>
Produtos Resultantes
<ul style="list-style-type: none"> <li>• Diagrama de Atividades (visão preliminar).</li> </ul>
Recursos
<p>Analista de Sistemas.</p> <p>Usuário.</p> <p>Computador.</p> <p>Software de diagramação (ex. Rational Rose, Jude, Argo UML).</p>

Quadro 9: Elaborar o Diagrama de Atividades.

Como suporte ao *workflow*, também foi elaborado um mapa conceitual do diagrama em questão (Figura 3.7), visando a agregar valor ao cognitivo dos alunos, ou seja, conceitos sem os quais o aluno pode ter maior dificuldade elaborar tais diagramas. Para todos os conceitos do mapa conceitual, foi elaborado uma tabela com a definição e notação conforme o Apêndice B.

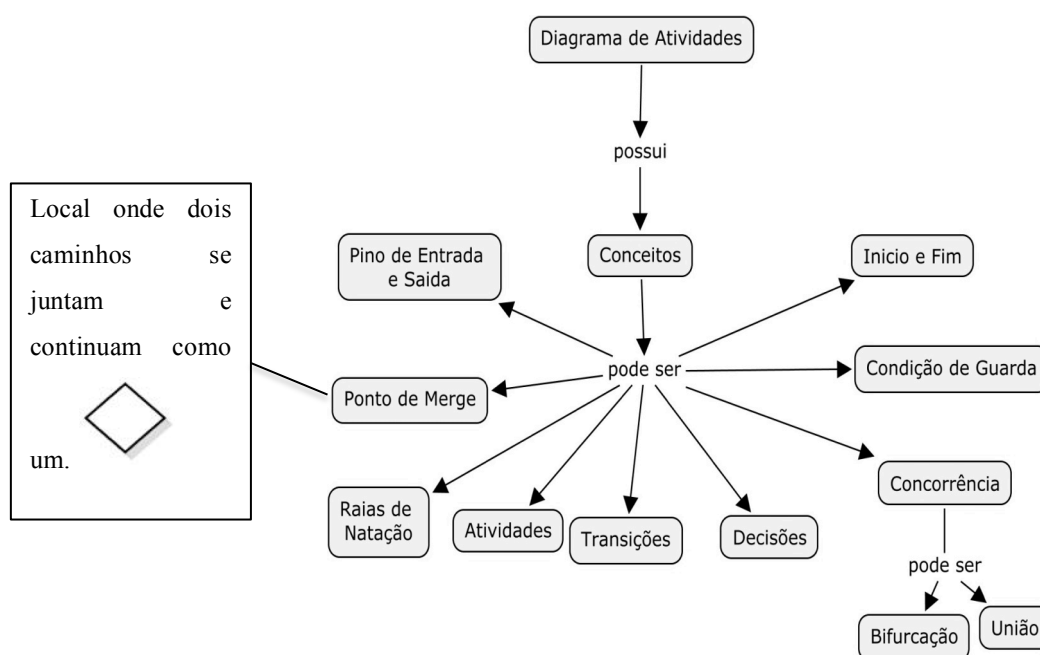


Figura 3.7: Mapa Conceitual do Diagrama de Atividades.

### 3.4 MAPA CONCEITUAL E *WORKFLOW* DO DIAGRAMA DE CLASSE

Os diagramas de classes são os diagramas encontrados com maior frequência na modelagem de sistemas orientados a objetos. Um diagrama de classe mostra um conjunto de classe, interfaces e colaborações e seus relacionamentos [3].

Uma classe representa um conjunto de objetos, características e comportamentos. A ela são atribuídos objetos com características semelhantes, sendo que ela define o comportamento de seus objetos através de métodos (ações) e quais estados eles são capazes de conter, por meio de atributos (características).

O *workflow* e o mapa conceitual para o diagrama de classe foram elaborados com o objetivo de auxiliar o aluno no aprendizado.



O *workflow* exibido nas Figuras 3.8, 3.9 e 3.10 demonstra os passos para a elaboração do diagrama de classe e o mapa conceitual da Figura 3.11, demonstra os conceitos como forma facilitadora de uma aprendizagem significativa.

Como pode-se verificar na Figura 3.8, a elaboração do diagrama de classe inicia-se com uma decisão em relação à compreensão dos conceitos relacionados com o diagrama de classe ou não. Se a decisão for “Sim”, será exibido os mapas conceituais referente ao diagrama de classe, caso contrário iniciará com o levantamento das possíveis classes do sistema, posteriormente têm-se duas atividades que acontecem paralelamente: a definição dos atributos e a definição das operações.

Ao término das duas atividades, iniciam-se outras duas atividades paralelas: a atividade estabelecer associação e definir a multiplicidade. Para o entendimento do problema e para agregar qualidade ao desenvolvimento do software, é necessária a realização de refinamentos sucessivos. Isto é representado no *workflow* pelo *fork* condicional, que acontece após a execução das duas atividades anteriores, retornando ou não para o início do *workflow*.

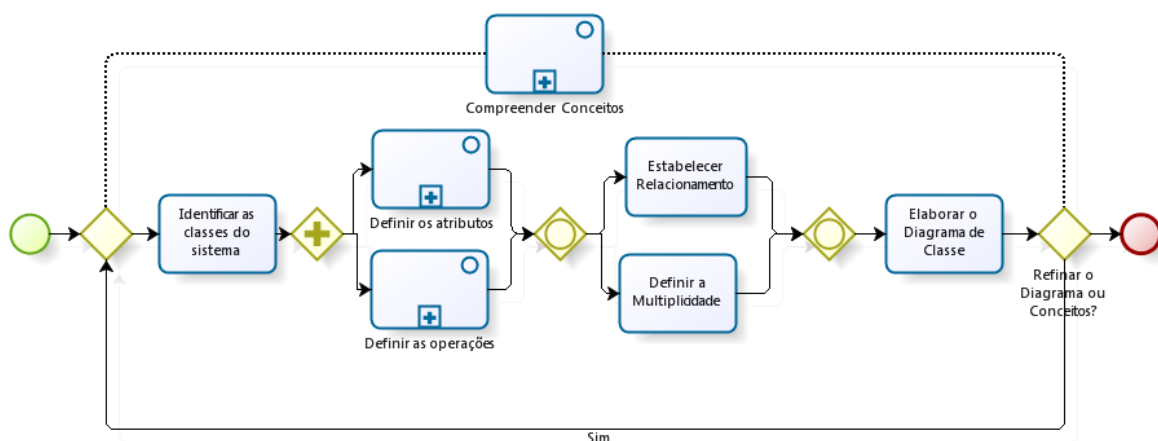


Figura 3.8: *Workflow* do Diagrama de Classe.

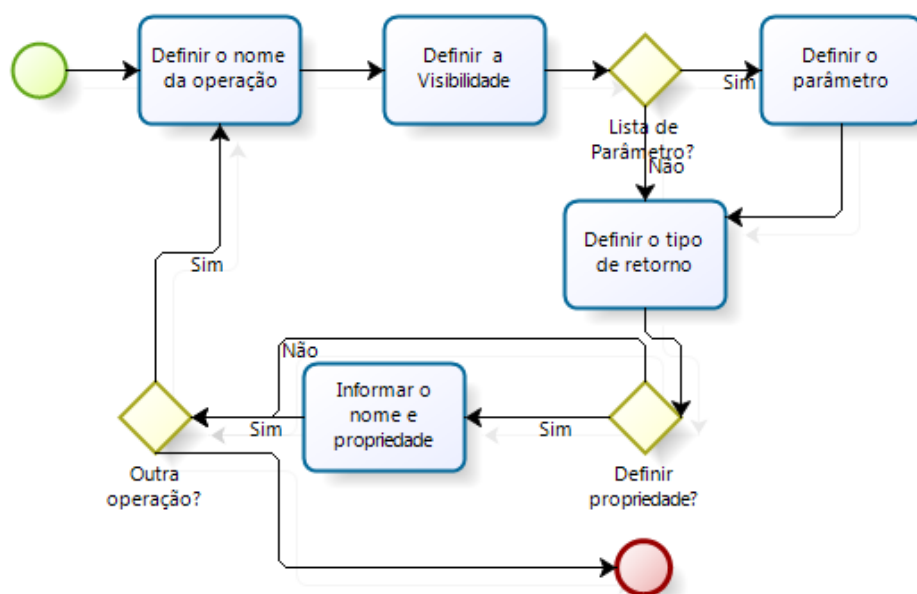


Figura 3.9: *Workflow* da Atividade Definir Operações (Diagrama de Classe).

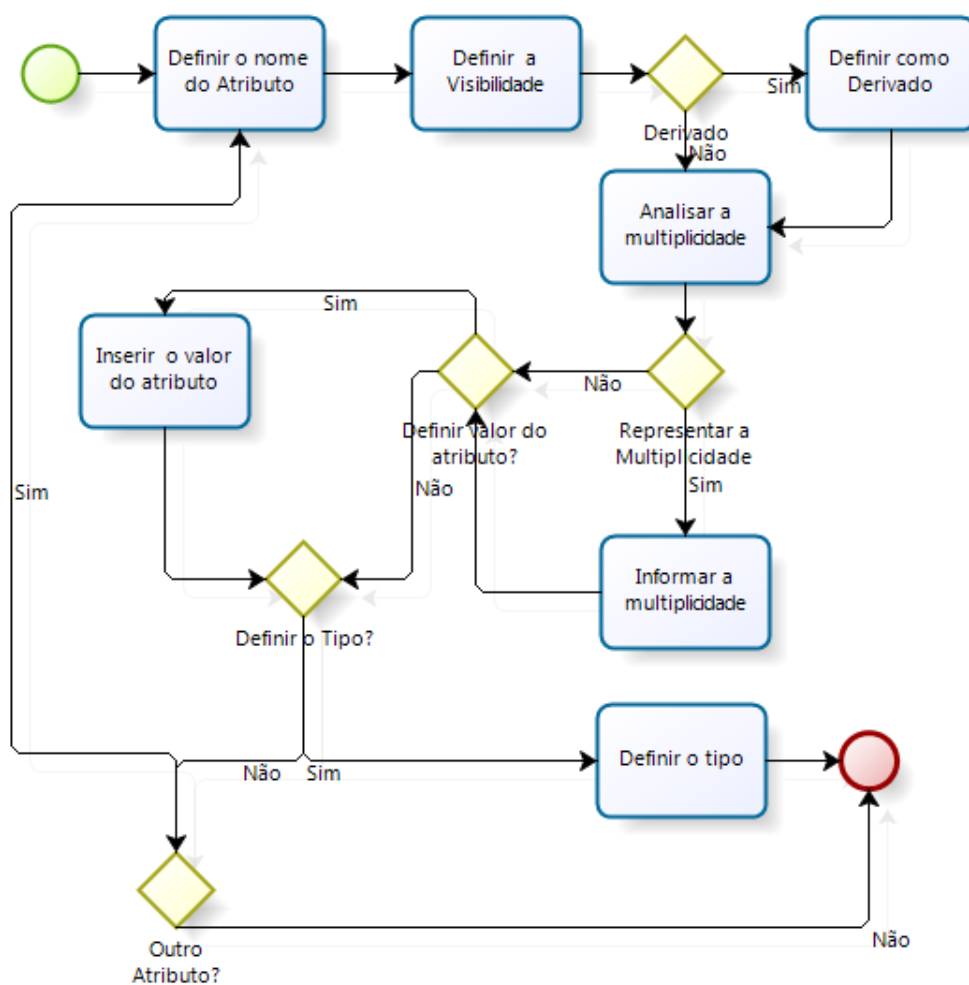


Figura 3.10: *Workflow* da Atividade Definir Atributo (Diagrama de Classes)

Várias atividades compõem o *workflow* do diagrama de classe. Cada atividade é composta por diversos elementos, sendo que estes estão detalhados nos Quadros 10, 11, 12, 13, 14 e 15 respectivamente.

<p>Todas as classes devem fazer sentido no domínio da aplicação. Nem todas as classes surgem explicitamente na definição do problema; algumas estão implícitas no domínio da aplicação ou no conhecimento geral [30].</p> <p>A partir da definição do requisitos, comece a listar as classes de objetos candidatas encontradas na descrição do problema. Não seja muito seletivo; registre todas as classes que vierem à cabeça. As classes muitas vezes correspondem a substantivos. Por exemplo, na frase “um sistema de reservas para vender entradas para peças em diversos teatros”, as classes possíveis seriam “Reserva, Sistema, Entrada, Peça e Teatro” [30].</p>
Metodologia
Conhecimento do negócio.
Produtos das Fases Anteriores
Levantamento de requisitos.
Produtos Resultantes
<ul style="list-style-type: none"> <li>• Diagrama de Classe (visão preliminar).</li> </ul>
Recursos
<p>Analista de Sistemas.</p> <p>Usuário.</p> <p>Computador.</p> <p>Software de diagramação (ex. Rational Rose, Jude, ArgoUML).</p>

Quadro 10: Identificar as Classes do Sistema.

<p>Um atributo é uma propriedade nomeada de uma classe que descreve um intervalo de valores que as instâncias da propriedade podem apresentar [3].</p> <p>Conforme Rup [19] , durante a definição de <b>métodos</b> e a identificação de <b>estados</b>, são identificados os atributos que a classe precisa para executar suas operações. Os atributos fornecem armazenamento de informações para a instância da classe e muitas vezes são utilizados para representar o estado da instância da classe. Qualquer informação que a própria classe mantém, ela o faz através de seus <b>atributos</b>. Para cada atributo, defina:</p>
---

<ul style="list-style-type: none"> <li>• seu <b>nome</b>, que deve obedecer às convenções de nomenclatura da linguagem de implementação e do projeto;</li> <li>• seu <b>tipo</b>, que será um tipo de dado elementar suportado pela linguagem de implementação;</li> <li>• seu <b>valor padrão ou inicial</b>, com o qual é inicializado quando novas instâncias da classe são criadas;</li> <li>• a <b>visibilidade</b>, que irá adotar um dos seguintes valores: <ul style="list-style-type: none"> <li>• <b>público</b>: o atributo é visível dentro e fora da pacote que contém a classe;</li> <li>• <b>protegido</b>: o atributo é visível somente para a própria classe, para suas subclasses ou para amigos da classe (dependente da linguagem);</li> <li>• <b>privado</b>: o atributo é visível somente para a própria classe e para amigos da classe;</li> <li>• <b>pacote</b>: o atributo é visível somente para a próprio pacote.</li> </ul> </li> <li>• a <b>multiplicidade</b> especifica a quantidade valores que podem estar associados a um elemento do modelo.</li> <li>• <b>String_propriedade</b> O elemento propriedades permite acrescentar praticamente qualquer informação adicional sobre o atributo que não se encaixe em um dos elementos predefinidos [1].</li> </ul>
Metodologia
Reunião com os envolvidos.
Produtos das Fases Anteriores
Levantamento de requisitos.
Produtos Resultantes
<ul style="list-style-type: none"> <li>• Classe com os atributos definidos.</li> </ul>
Recursos
Analista de Sistemas. Usuário. Computador. Software de diagramação (ex. Rational Rose, Jude, ArgoUML).

Quadro 11: Definir Atributos

Uma operação é uma abstração de algo que pode ser feito com um objeto e que é compartilhado por todos os objetos dessa classe. Uma classe pode ter qualquer número de operações ou até nenhuma operação [3].

Para identificar operações em classes [19]:

- Estude as responsabilidades de cada classe de análise correspondente, criando uma operação para cada responsabilidade. Use a descrição da responsabilidade como a descrição inicial da operação.
- Estude as realizações de casos de uso nas participações de classe para ver como as operações são utilizadas pelas realizações de casos de uso. Amplie as operações, uma realização de Caso de Uso por vez, refinando as operações, suas descrições, tipos de retorno e parâmetros. Os requisitos de cada realização de Caso de Uso pertencentes às classes são descritos textualmente no Fluxo de Eventos da realização de Caso de Uso.
- Analise o Caso de Uso Requisitos Especiais para certificar-se de que não falta nenhum requisito implícito na operação que poderá ser declarada.

A operação possui a sintaxe:

[visibilidade] nome ([lista-parâmetro]) “:” [ resultado-retorno] [(propriedades)]

lista-parâmetro := nome [ “:” tipo-dado] [multiplicidade]

#### Definindo a Visibilidade da Operação

Para cada operação, identifique a visibilidade de exportação da operação dentre estas opções [19]:

- **pública** - a operação é visível para modelar elementos que não sejam a própria classe;
- **pacote** - a operação é visível somente dentro do próprio pacote;
- **protegida** - a operação é visível somente para a própria classe, para suas subclasses ou para *amigos* da classe (dependente de linguagem);
- **privada** - a operação é visível somente para a própria classe e para *amigos* da classe.

#### Nomeando e Descrevendo as Operações

Utilize as convenções de nomenclatura da linguagem de implementação, quando estiver

nomeando operações, tipos de retorno, bem como parâmetros e seus tipos [19].

#### Lista-parâmetro

A lista-parâmetro é uma lista ordenada dos atributos, que, juntos, definem a entrada para uma operação [1].

#### Resultado-retorno

O resultado-retorno é a saída da operação [1].

#### Propriedades

O elemento propriedades permite acrescentar praticamente qualquer informação adicional sobre a operação que não se encaixe em um dos elementos predefinidos [1].

#### Metodologia

Reunião com os envolvidos.

#### Produtos das Fases Anteriores

Levantamento de requisitos.

#### Produtos Resultantes

- Diagrama de Classe.
- Operação da Classe.

#### Recursos

Analista de Sistemas.

Usuário.

Computador.

Software de diagramação (ex. Rational Rose, Jude, Argo UML).

Quadro 12: Definir Operação

As associações proporcionam o mecanismo para os objetos se comunicarem entre si. Elas fornecem aos objetos um canal pelo qual as mensagens podem fluir. Além disso, documentam as dependências entre as classes, destacando que as mudanças em uma classe podem ser percebidas entre muitas outras classes [19].

#### Metodologia

Reunião.

#### Produtos das Fases Anteriores

Levantamento de requisitos.
Produtos Resultantes
• Diagrama de Classe.
Recursos
Analista de Sistemas. Usuário. Computador. Software de diagramação (ex. Rational Rose, Jude, Argo UML).

Quadro 13: Estabelecer Relacionamento

A multiplicidade procura determinar o número mínimo e o máximo de objetos envolvidos em cada extremidade da associação, além de permitir especificar o nível de dependência de um objeto para com os outros envolvidos na associação [28].
Metodologia
Reunião com os envolvidos.
Produtos das Fases Anteriores
Levantamento de requisitos.
Produtos Resultantes
• Diagrama de Classe com a multiplicidade.
Recursos
Analista de Sistemas. Usuário. Computador. Software de diagramação (ex. Rational Rose, Jude, Argo UML).

Quadro 14: Definir a Multiplicidade.

Os diagramas de classes mostram a estrutura estática do modelo, principalmente os elementos existentes, como classes, sua estrutura interna e seus relacionamentos com outras classes. Eles não mostram informações temporárias [20].  Um diagrama de classes é apresentado como um conjunto de elementos do modelo declarativos (estáticos) - como classes, pacotes e seus relacionamentos - que são conectados entre si e a seu conteúdo como um gráfico. Os diagramas de classe podem ser organizados em (e pertencentes a) pacotes, mostrando apenas o que é relevante em um determinado pacote [20].
---

Metodologia
Reunião com os envolvidos.
Produtos das Fases Anteriores
Diagrama de Caso de Uso. Especificação de Caso de Uso.
Produtos Resultantes
• Diagrama de Classe
Recursos
Analista de Sistemas. Usuário. Computador. Software de diagramação (ex. Rational Rose, Jude, Argo UML).

Quadro 15: Elaborar o Diagrama de Classe.

Como suporte ao *workflow*, foi elaborado um mapa conceitual do diagrama de classe, conforme demonstrado na Figura 3.11, onde o aluno pode encontrar os conceitos envolvidos para elaboração do diagrama. A descrição dos conceitos estão representados no Apêndice B.



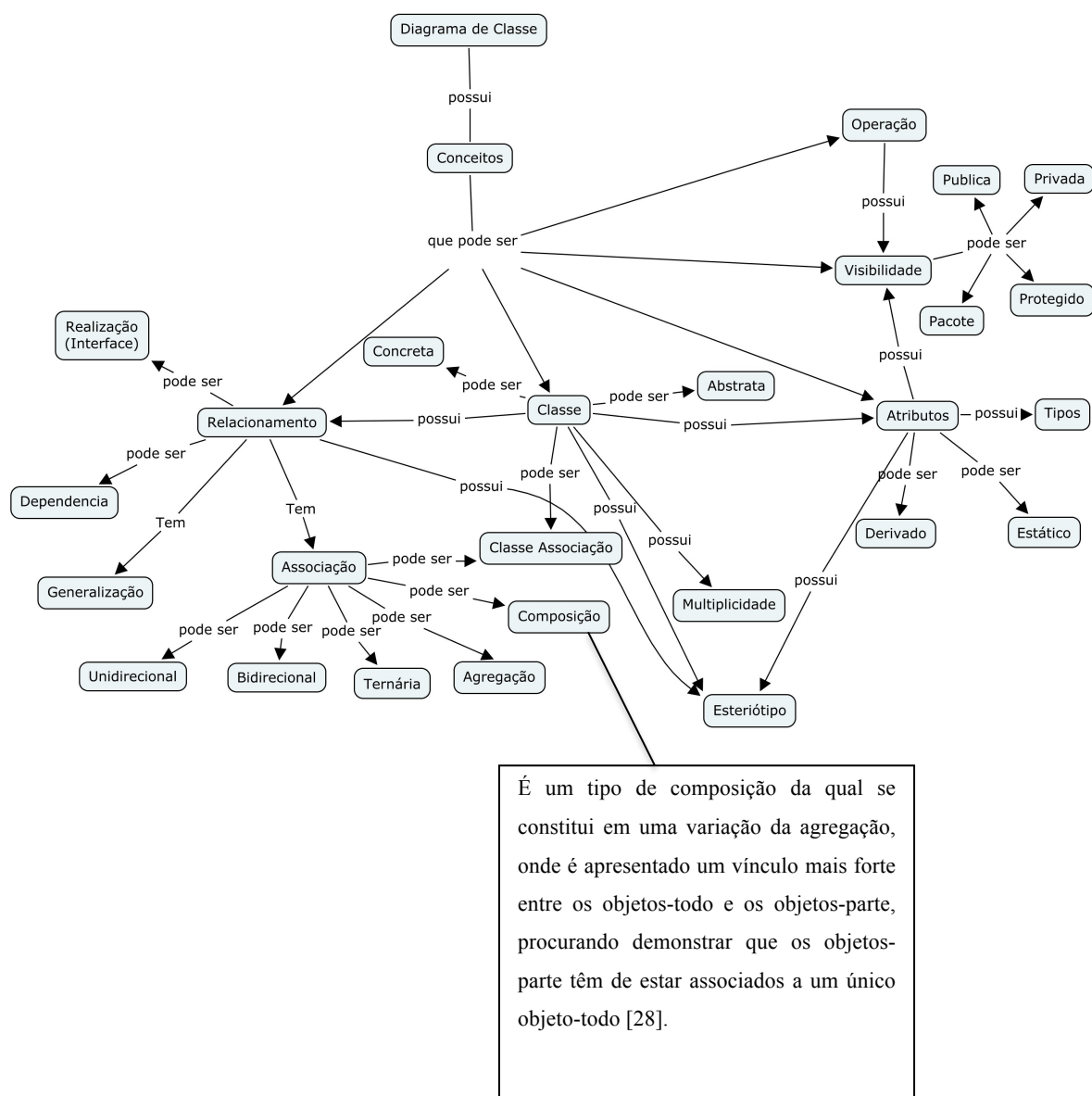


Figura 3.11: Mapa Conceitual do Diagrama de Classe.

### 3.5 MAPA CONCEITUAL E *WORKFLOW* DO DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência é um diagrama comportamental que procura determinar a sequência de eventos que ocorrem em um determinado processo, identificando quais mensagens devem ser disparadas entre os elementos envolvidos e em que ordem ocorrem [28]. Dessa forma, o diagrama em questão mostra a interação entre os objetos dentro de um determinado processo.

O diagrama de sequência baseia-se no diagrama de caso de uso, onde normalmente existe um diagrama de sequência para cada caso de uso. O diagrama de sequência depende também do diagrama de classe [28].

O *workflow* demonstrado na Figura 3.12 e Figura 3.13 demonstra os passos para elaboração do diagrama de sequência, e o mapa conceitual da Figura 3.14 demonstra os conceitos como forma facilitadora de uma aprendizagem significativa.

Conforme apresentado na Figura 3.12, a elaboração do diagrama de sequência inicia-se com uma decisão em relação à compreensão dos conceitos relacionados com o diagrama de sequência ou não. Se a decisão for “Sim”, será exibido os mapas conceituais referente ao diagrama de sequência, caso contrário iniciará pela escolha do caso de uso para o qual será efetuado o diagrama de sequência.

Posteriormente, têm-se duas atividades que acontecem paralelamente: a alocação de atores, onde se deve definir os atores que fazem parte do cenário, bem como os objetos. Ao término das duas atividades, inicia-se o diagrama de sequência, buscando os eventos e métodos. Durante a elaboração do diagrama de sequência, pode ser necessário realizar refinamentos sucessivos, com o objetivo de melhorá-lo. Isto é representado no *workflow* pelo *fork* condicional, que acontece após a execução das duas atividades anteriores, retornando ou não para o início do *workflow*.

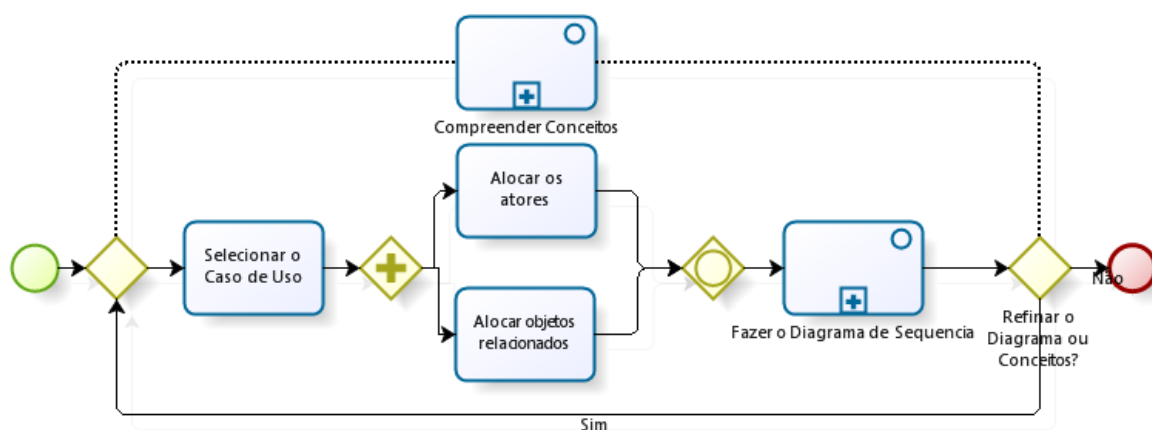


Figura 3.12: *Workflow* do Diagrama de Sequência.

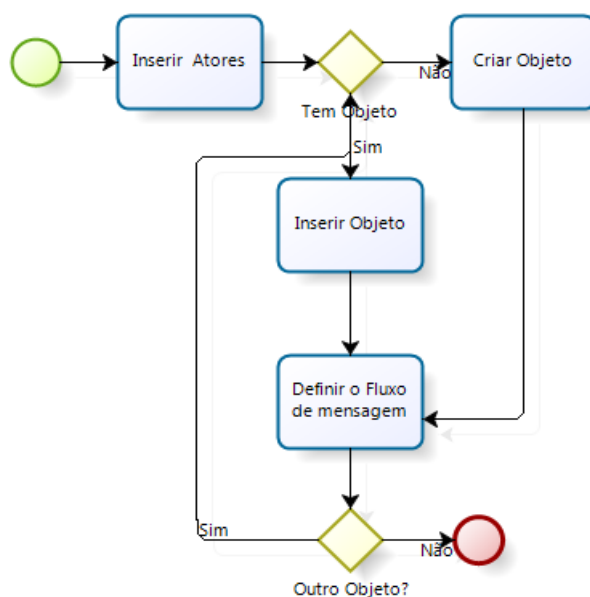


Figura 3.13: *Workflow* da Atividade “Fazer Diagrama de Sequência” (Diagrama de Sequência).

Várias atividades compõem o *workflow* do diagrama de sequencia. O Cada atividade é composta por diversos elementos, sendo que estes estão detalhados nos Quadros 16, 17, 18 e 19 respectivamente.

A atividade selecionar Caso de Uso consiste em escolher para qual Caso de Uso será efetuado o diagrama de sequência.
Metodologia
Reunião com a equipe técnica.
Produtos das Fases Anteriores
Diagrama de Caso de Uso. Diagrama de Classe.
Produtos Resultantes
Diagrama de Sequência visão preliminar.
Recursos
Analista de Sistemas. Usuário. Computador. Software de diagramação (ex. Rational Rose, Jude, ArgoUML).

Quadro 16: Seleciona o Caso de Uso.

<p>A atividade alocar atores é destinada a buscar os atores já existentes e/ou criá-los, se necessário.</p> <p>Normalmente, os atores já foram identificados, ao se efetuar o Diagrama de Caso de Uso.</p> <p>Apenas os atores que vão interagir farão parte do diagrama.</p>
Metodologia
Reunião com a equipe técnica.
Produtos das Fases Anteriores
<p>Diagrama de Caso de Uso.</p> <p>Identificação dos atores envolvidos.</p> <p>Diagrama de Classe.</p>
Produtos Resultantes
Diagrama de Sequência visão preliminar.
Recursos
<p>Analista de Sistemas.</p> <p>Usuário.</p> <p>Computador.</p> <p>Software de diagramação (ex. Rational Rose, Jude, ArgoUML).</p>

Quadro 17: Alocar Atores.

<p>A atividade “Alocar objetos” é destinada a buscar os objetos já existentes e/ou criá-los, se necessário.</p> <p>Se o diagrama de classe já foi elaborado, a grande parte dos objetos já está criada.</p>
Metodologia
Reunião com a equipe técnica.
Produtos das Fases Anteriores
<p>Diagrama de Caso de Uso</p> <p>Identificação dos atores envolvidos.</p> <p>Diagrama de Classe.</p>
Produtos Resultantes
<ul style="list-style-type: none"> <li>• Diagrama de Sequência visão preliminar.</li> </ul>
Recursos
Analista de Sistemas.

Usuário. Computador. Software de diagramação (ex. Rational Rose, Jude, ArgoUML).
--

Quadro 18: Alocar os Objetos Relacionados.

<p>A atividade “Fazer diagrama de sequência” é onde o diagrama torna-se completo. Nesta atividade, os objetos já existentes são utilizados.</p>
---

<p>A partir do instante em que os atores que vão interagir no diagrama já estão definidos, os objetos já foram criados quase em sua totalidade, a elaboração do diagrama de sequência torna-se simples.</p>
---

<p>Dando sequência, basta colocar os fluxos de mensagens, definir qual o tipo de mensagem e suas propriedades (seus parâmetros), bem como as operações.</p>
---

Metodologia
-------------

Reunião com a equipe técnica.
-------------------------------

Produtos das Fases Anteriores
-------------------------------

Diagrama de Caso de Uso.
--------------------------

Identificar os atores envolvidos.
-----------------------------------

Diagrama de Classe.
---------------------

Produtos Resultantes
----------------------

- |   |
|---|
| <ul style="list-style-type: none"> <li>• Diagrama de Sequência completo.</li> </ul> |
|---|

Recursos
----------

Analista de Sistemas.
-----------------------

Usuário.
----------

Computador.
-------------

Software de diagramação (ex. Rational Rose, Jude, ArgoUML).
---

Quadro 19: Fazer o Diagrama de Sequência.

Como suporte ao *workflow*, foi elaborado um mapa conceitual do diagrama de sequência, conforme demonstrado na Figura 3.14, onde o aluno pode encontrar os conceitos envolvidos para elaboração do diagrama. A descrição dos conceitos estão representados no Apêndice B.

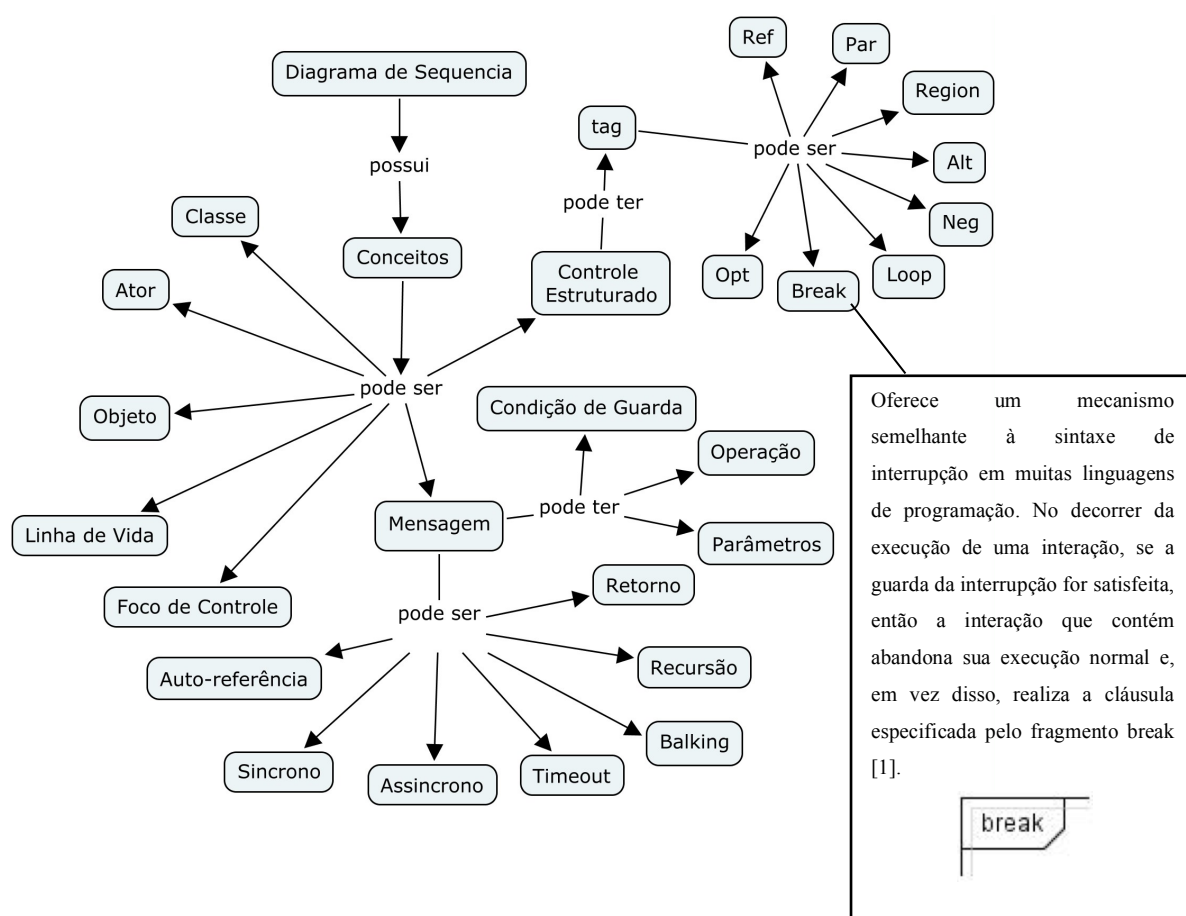


Figura 3.14: Mapa Conceitual do Diagrama de Sequência.

### 3.6 CONSIDERAÇÕES FINAIS

Neste Capítulo foi possível visualizar os mapas conceituais e *workflow* de cada diagrama trabalhado no âmbito do trabalho.

Através das atividades do *workflow* é possível explorar os mapas conceituais. Cada conceito do mapa conceitual está detalhado no Apêndice B.

Cada atividade do *workflow* é composta pelas instruções de trabalho, tornando-o ainda mais completo.

## 4 ESTUDO DE CASO

Para verificar e validar a aplicação do *workflow* e dos mapas conceituais no processo de ensino e aprendizagem da UML, foi realizada uma pesquisa, que consiste na elaboração e realização de aulas. Nelas foram apresentados e aplicados o *workflow* e os mapas conceituais, desenvolvidos no âmbito desta pesquisa, a alunos que já detinham ou não algum conhecimento da UML. Isso foi feito com o propósito de avaliar o grau de contribuição do *workflow* e do mapa conceitual, a fim de se ter um indicativo da eficácia da solução apresentada.

Segundo Menestrina e Goudard [23], é muito importante a maneira como o docente apresenta o material (diretamente, mediante livros ou outros materiais didáticos), e que esta apresentação consiga atualizar a significação intrínseca de conteúdo em uma situação concreta e efetiva, que facilite e ative os esquemas de conhecimentos pertinentes e que estimule e incentive os alunos, modificando uma disposição (por vez) desfavorável em favorável para realizar a aprendizagem significativa.

A metodologia utilizada para ministrar as aulas foi a preleção ou aula expositiva, com a participação dos alunos (dialogada). O conteúdo foi trazido pelo professor inteiramente delimitado, justamente por se buscarem conclusões sobre o uso dos mapas conceituais e *workflow* no processo de ensino e aprendizagem. O material foi inserido no aplicativo Power Point da Microsoft. Como recursos de apoio foram utilizados um computador e um projetor multimídia (*datashow*). Como apoio, o instrumento foi disponibilizado no endereço [http://www.gaia.uel.br/oa\\_uml/Index.html](http://www.gaia.uel.br/oa_uml/Index.html).

As aulas relativas ao contexto deste estudo de caso foram conduzidas de modo diferente a da ordem do conteúdo ministrado, para averiguar a forma mais favorável à aprendizagem dos alunos. Elas foram ministradas na Universidade Estadual de Londrina (UEL), a alunos do Curso de Ciência da Computação e aos alunos do projeto de pesquisa da fábrica de software GAIA e no Centro Universitário Filadélfia de Londrina (UniFil), a alunos de Graduação em Ciência da Computação e Sistemas de Informação.

Todas as aulas foram iniciadas dando ênfase ao *workflow*, mapa conceitual e à hierarquia de construção dos diagramas. Quanto à exposição do conteúdo específico de cada diagrama, a sequência do conteúdo aconteceu de forma diferenciada.

As aulas foram ministradas em diferentes instituições de ensino, com um variado número de alunos, conforme demonstrado no Quadro 20.

Diagrama	Quantidade			Curso	Instituição de Ensino
	Turmas	Total de Alunos	Slides		
Caso de Uso	3	60	53	Sistemas de Informação Ciência da Computação Projeto Pesquisa Gaia	UniFil UEL
Atividades	2	36	53	Sistemas de Informação Ciência da Computação	UniFil UEL
Classe	1	25	36	Ciência da Computação	UniFil
Sequência	3	44	39	Sistemas de Informação Ciência da Computação	UniFil

Quadro 20: Relação das Aulas Ministradas

Conforme o Quadro 20, o diagrama de caso de uso foi ministrado em 3 turmas, totalizando 60 alunos, sendo eles dos cursos de Sistemas de Informação e Ciência da Computação de duas instituições de ensino superior, UniFil e UEL e aos alunos do projeto de pesquisa Gaia. O diagrama de atividades, foi ministrado em 2 turmas, totalizando 36 alunos, dos cursos de Sistemas de Informação e Ciência da Computação da UniFil e UEL. O Diagrama de Classe foi ministrado em apenas uma turma com 25 alunos da UniFil e o diagrama de sequência foi ministrado em duas turmas, totalizando 44 alunos dos cursos de Sistemas de Informação e Ciência da Computação da UniFil.

A forma com que as aulas foram apresentadas aos alunos teve influência na assimilação do conteúdo. Segue abaixo as características de cada aula:

- diagrama de caso de uso: foram iniciadas pela explicação do *workflow* para a elaboração do diagrama de caso de uso e suas instruções de trabalho. Explicado o *workflow*, deu-se prosseguimento à explicação do mapa conceitual referente à elaboração do diagrama de caso de uso. Cada aula teve a duração de aproximadamente 1 hora e 15 minutos, e, em seu decorrer, foi dada a palavra aos alunos, para emitirem comentários ou sanarem dúvidas;



- diagrama de atividades: para este diagrama, a aula foi ministrada de forma diferenciada. A aula foi iniciada com explicação do que é o diagrama de atividades e a demonstração de uma imagem do diagrama, para que os alunos tivessem uma visão macro do assunto que seria abordado em sala. Somente após situar o aluno, foi demonstrado o mapa conceitual do diagrama de atividades e foram abordados todos os conceitos relevantes. Explicado o mapa conceitual, iniciou-se a explicação do *workflow* do diagrama de atividades e do *workflow* da supertarefa “Elaborar Diagrama de Atividades”. A aula teve a duração de aproximadamente uma hora;
- diagrama de classe: foi ministrado seguindo a mesma metodologia aplicada pelo diagrama de atividades;
- diagrama de sequencia: foi ministrado seguindo a mesma metodologia aplicada pelo diagrama de atividades.

Analisando somente a sequência de demonstração da aula aos alunos, foi perceptível a diferença quanto à assimilação do conteúdo. Situar o aluno em relação ao assunto abordado, permitiu aproveitamento significativo no conteúdo.

O instrumento de coleta de dados que foi utilizado para o desenvolvimento da pesquisa em relação a todos os diagramas foi um questionário, que, por suas características próprias, tem a vantagem da rapidez e poder incluir a opinião de todos os alunos que participaram da aula. Ao término de cada aula, foi solicitado aos alunos que preenchessem o questionário, composto de oito perguntas (Apêndice A), visando a demonstrar o grau de contribuição que o *workflow* e os mapas conceituais lhe trouxeram.

Para análise dos resultados, foi aplicada a análise estatística do qui quadrado, utilizando-se o quadro estatístico de valores mínimos em um teste de diferença direcional bi-lateral [33] cujos resultados foram considerados como “sim” e “não” em uma primeira etapa e, em seguida, aqueles com respostas positivas, convertidos em porcentagens de categorias: “Sem opinião”, “Não”, “Totalmente” e “Parcialmente”.

## 4.1 RESULTADOS

Para verificar a contribuição do *workflow* e dos mapas conceituais na forma de estudar e aprender a elaborar os diagramas das UML em questão, foram aplicados questionários (Apêndice A), como já citado.

No que diz respeito à análise dos questionários, os dados foram tabulados e, a partir destes, foram gerados gráficos, a fim de melhorar a visualização dos resultados, conforme expostos nas seções seguintes.

### 4.1.1 Diagrama de Caso de Uso

Ao questionar os alunos que tiveram contato com a UML, quanto ao conhecimento agregado com o instrumento aplicado, de 60 alunos pesquisados, 67% responderam que tiveram 50% ou mais conhecimento agregado, sendo que para apenas 33% a resposta foi entre 0% a 40%, de acordo com o gráfico da Figura 4.1.

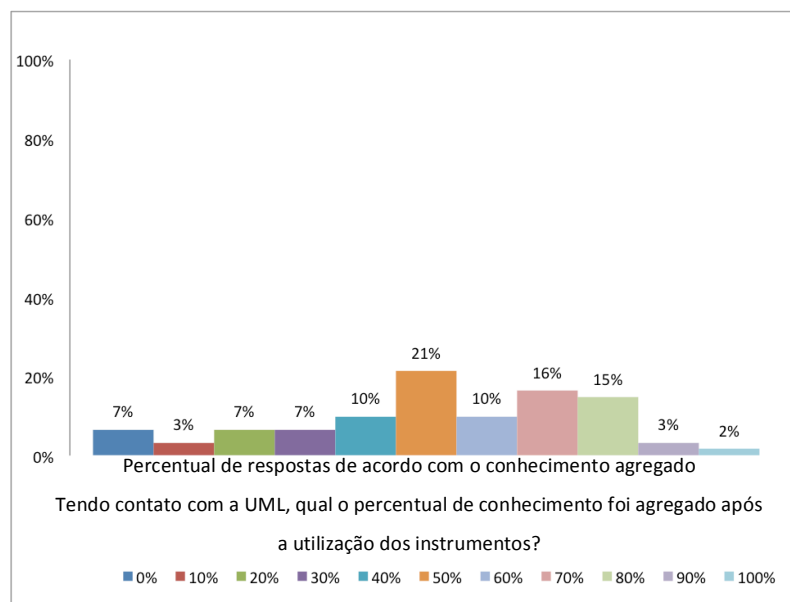


Figura 4.1: Histograma da questão 2 - Diagrama de Caso de Uso.

Mesmo com algum conhecimento em relação ao diagrama em questão, a pesquisa demonstrou que o instrumento foi importante para um melhor entendimento do assunto.

Quando questionado se “o *workflow* e as instruções de trabalho contribuem para o processo aprendizagem dos diagramas da UML”, 10% dos alunos responderam que “não” sabiam opinar; 54% dos alunos responderam “Parcialmente”, sendo que 36 % responderam “Totalmente”. Levando em consideração as respostas “Parcialmente e Totalmente”, pode-se concluir que aproximadamente 90% responderam de forma positiva, de acordo com a Figura 4.2, mostrando dessa forma que o *workflow* para elaboração do diagrama de caso de uso auxilia no processo de aprendizagem.

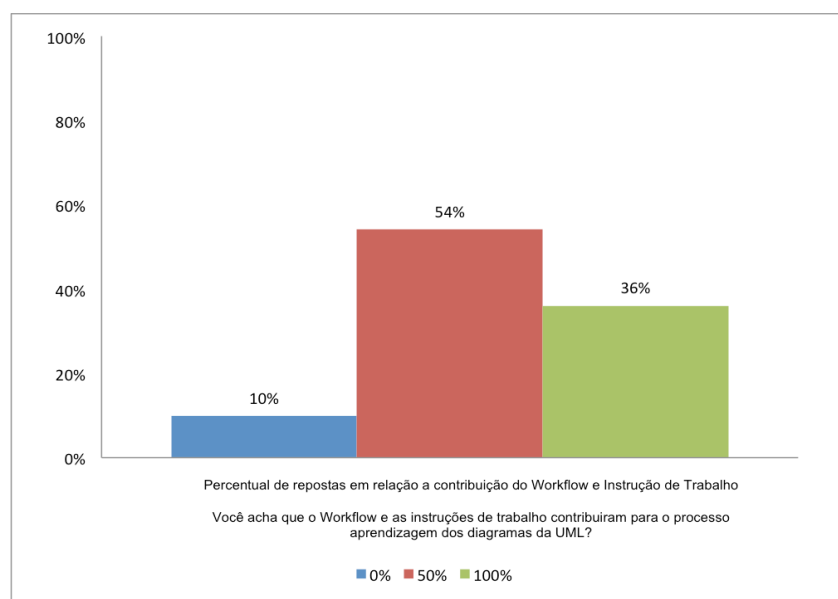


Figura 4.2: Histograma da questão 3 – Diagrama de Caso de Uso

Para cada atividade do *workflow* foi elaborado uma instrução de trabalho, que faz parte da atividade, de forma descritiva e completa. Ao questionar em relação a contribuição da instrução de trabalho juntamente com o *workflow*, os alunos pesquisados consideraram bastante importância, visto que somente 10% responderam que não sabiam opinar.

A mesma análise foi efetuada para a questão “O mapa conceitual tornou mais clara a forma de estudar os diagramas”, sendo que 10% dos alunos “não opinaram”, 7% dos alunos responderam que “não contribui” e 36% responderam “Parcialmente” e 48% responderam “Totalmente”, de acordo com a Figura 4.3.

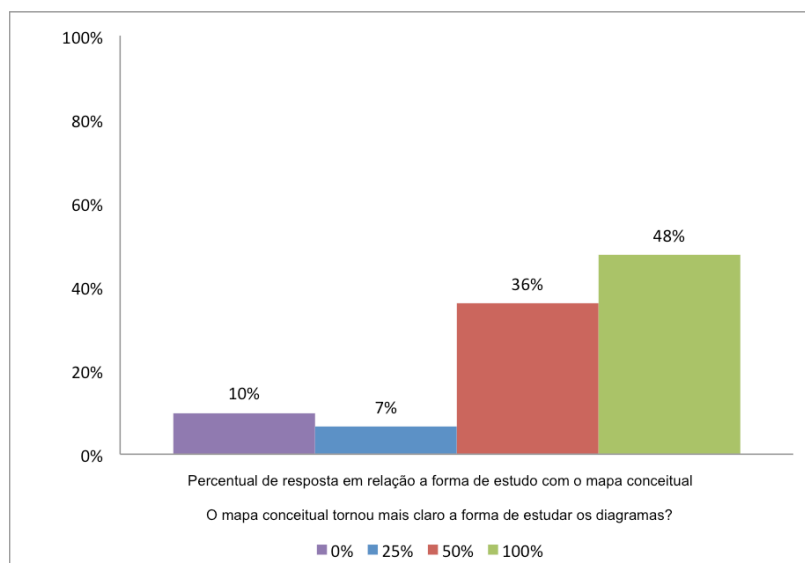


Figura 4.3: Histograma da questão 5 - Diagrama de Caso de Uso

Conforme a análise, o mapa conceitual apresentado foi significativo na aprendizagem do diagrama de caso de uso, onde 48% dos alunos questionados responderam que torna 100% mais claro a forma de estudar.

Quando analisado o conjunto, se o mapa conceitual e o *workflow* contribuíram para a aprendizagem, Figura 4.4, 8% responderam “Sem opinião”; 5% responderam “Não”; 51% responderam “Parcialmente” e 36% responderam “Totalmente”.

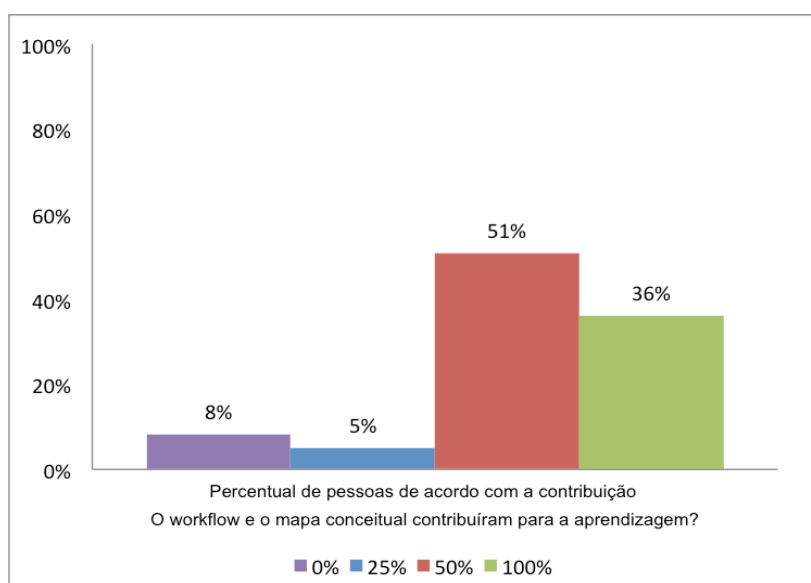


Figura 4.4: Histograma da questão 7- Diagrama de Caso de Uso.

A utilização dos dois instrumentos para o ensino e aprendizagem, foi satisfatória, onde, somando as pessoas que responderam acima de 50%, conclui-se que 87% dos alunos questionados responderam positivamente.

#### 4.1.2 Diagrama de Atividades

Similarmente ao diagrama de caso de uso, utilizou-se um questionário para avaliar o instrumento aplicado aos estudantes em relação ao diagrama de atividades.

Ao questionar os alunos que tiveram contato com a UML, quanto ao conhecimento agregado com o instrumento aplicado, de 36 alunos pesquisados, 70% responderam que tiveram 50% ou mais conhecimento agregado, sendo que para apenas 30% a resposta foi entre 0% a 40%, de acordo com o gráfico da Figura 4.5.

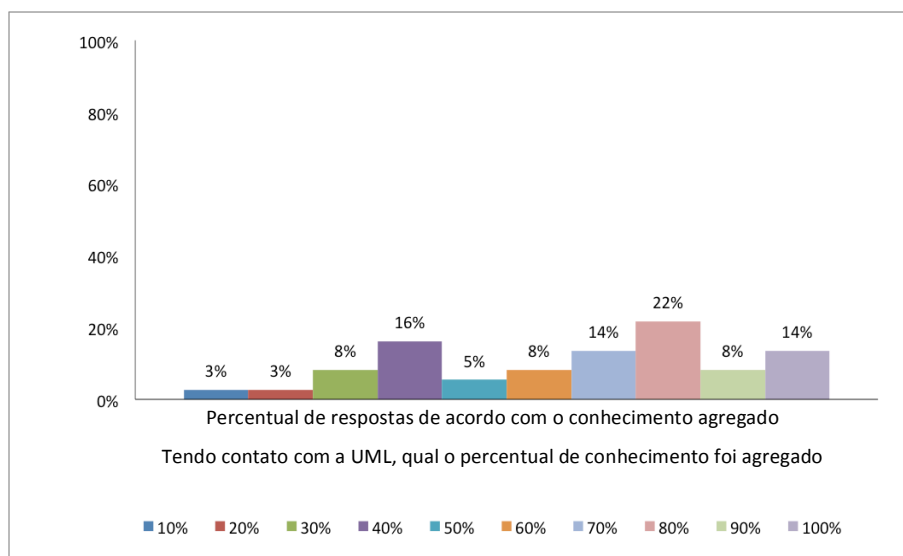


Figura 4.5: Histograma da questão 2 - Diagrama de Atividades.

Apesar do conhecimento em relação ao diagrama em questão, a pesquisa demonstrou que o instrumento foi importante para um melhor entendimento do assunto.

Quando questionado se “o *workflow* e as instruções de trabalho contribuem para o processo aprendizagem dos diagramas da UML”, 49% dos alunos responderam “Parcialmente”, sendo 54% responderam “Totalmente”. Nenhum aluno respondeu que “não contribui” ou “sem opinião”, conforme a Figura 4.6.

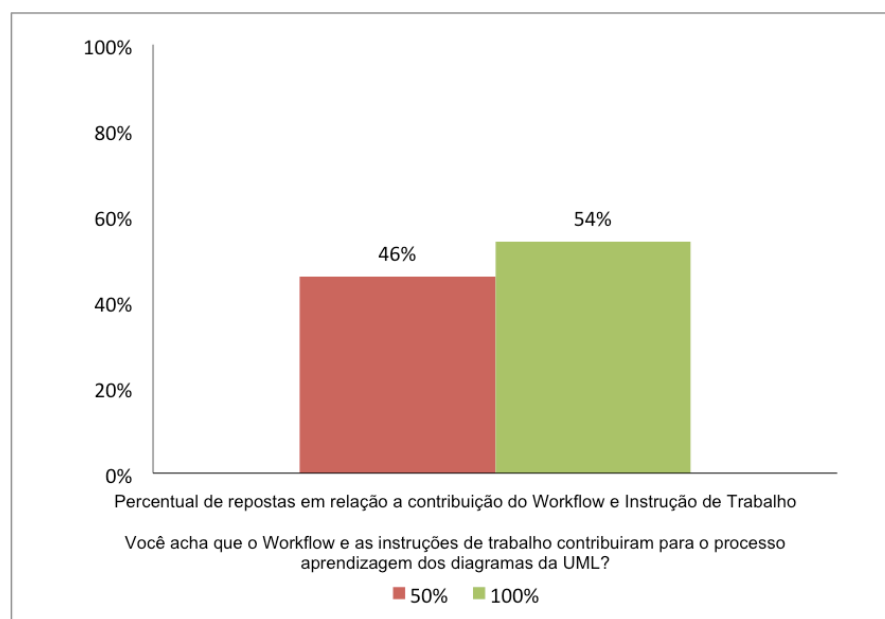


Figura 4.6: Histograma da questão 3 - Diagrama de Atividades.

O *workflow* juntamente com a instrução de trabalho tornou um instrumento importante no estudo do diagrama de atividades, de acordo o resultado apresentado na Figura 4.6.

A mesma análise foi efetuada para a questão “O mapa conceitual tornou mais claro a forma de estudar os diagramas”, 41% responderam “Parcialmente” e 59% responderam “Totalmente”, de acordo com a Figura 4.7. Assim como no questionamento anterior, não foi obtida nenhuma resposta indicativa de que o instrumento não tenha contribuído para aprendizagem.

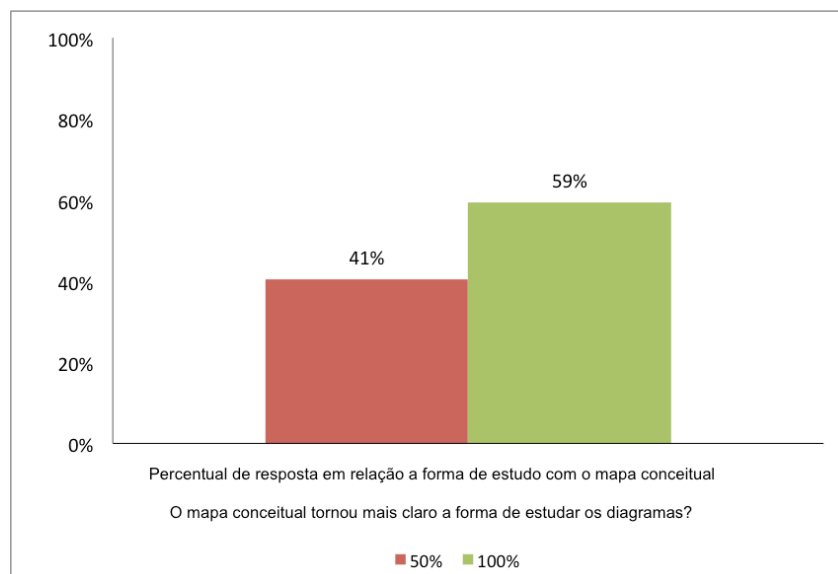


Figura 4.7: Histograma da questão 5 - Diagrama de Atividades.

Conforme a Figura 4.7, o mapa conceitual tornou-se um instrumento importante no estudo do diagrama de atividades. Através do mapa conceitual, os conceitos podem ser estudados e posteriormente aplicados na elaboração do diagrama.

Quando analisado o conjunto, indagando se o mapa conceitual e o *workflow* contribuíram para a aprendizagem, 41% responderam “Parcialmente” e 59% responderam “Totalmente”, conforme demonstrado na Figura 4.8.

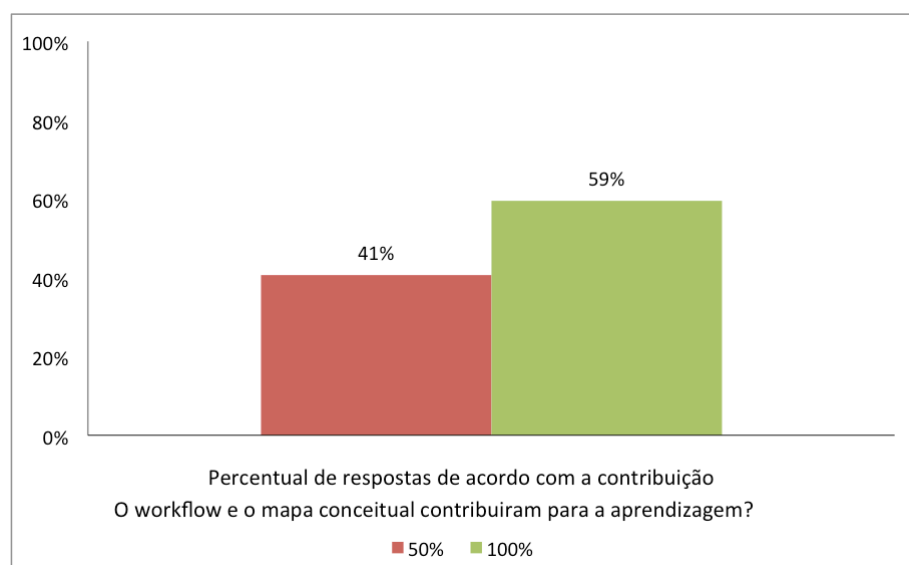


Figura 4.8: Histograma da questão 7 - Diagrama de Atividades.

A Figura 4.8 apresentou os resultados em relação aos dois instrumentos, e através dele é possível concluir que os alunos tiveram uma contribuição positiva e significativa.

#### 4.1.3 Diagrama de Classe

Após o conteúdo abordado em sala de aula, os alunos responderam um questionário (Apêndice A) como uma maneira para averiguação do conteúdo. Segue abaixo os resultados do questionário.

Ao questionar os alunos que tiveram contato com a UML, quanto ao conhecimento agregado com o instrumento aplicado, de 27 alunos pesquisados, 82%

responderam que tiveram 50% ou mais conhecimento agregado, sendo que para apenas 18% a resposta foi entre 0% a 40%, de acordo com o gráfico da Figura 4.9.

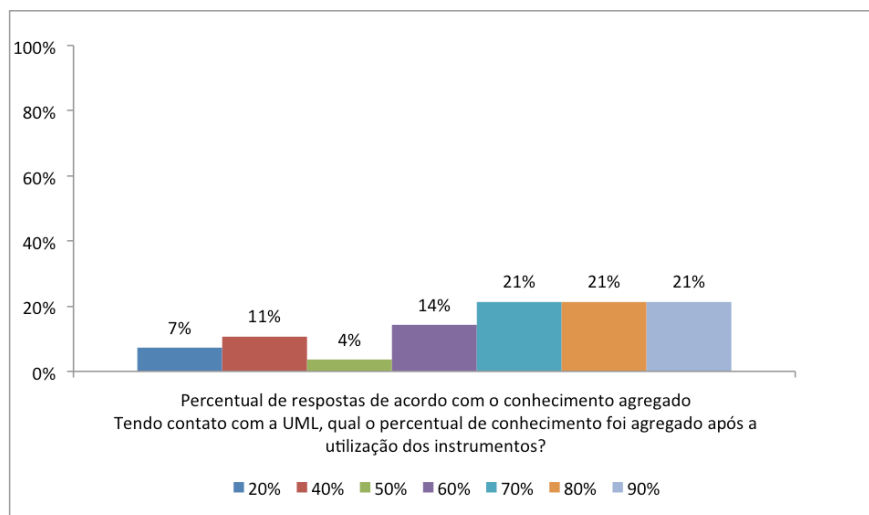


Figura 4.9: Histograma da questão 2 - Diagrama de Classe

O diagrama de classe é um dos principais diagramas da UML. A partir dele outros diagramas podem ser elaborados. É extremamente importante o seu entendimento e pela Figura 4.9 é possível verificar que mesmo com um conhecimento prévio, o instrumento foi importante para o estudo.

Quando questionado se “o *workflow* e as instruções de trabalho contribuem para o processo aprendizagem dos diagramas da UML”, 4% dos alunos responderam “Sem Opinião”, sendo 39% responderam “Parcialmente” e 57% responderam “Totalmente” conforme a Figura 4.10.



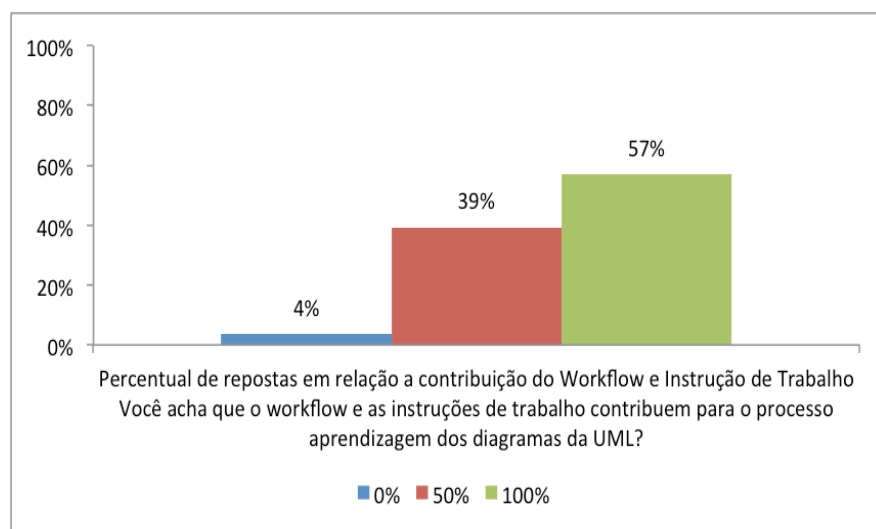


Figura 4.10: Histograma da questão 3 - Diagrama de Classe

A forma de visualizar melhor as atividades do *workflow* é complementada através das instruções de trabalho, sendo que, de acordo com a Figura 4.10, os alunos concluíram que obtiveram uma boa contribuição no estudo.

A mesma análise foi efetuada para a questão “O mapa conceitual tornou mais claro a forma de estudar os diagramas”, dos 27 questionados, 11% responderam “Sem Opinião”, 36% responderam “Parcialmente” e 54% responderam “Totalmente”, de acordo com a Figura 4.11.

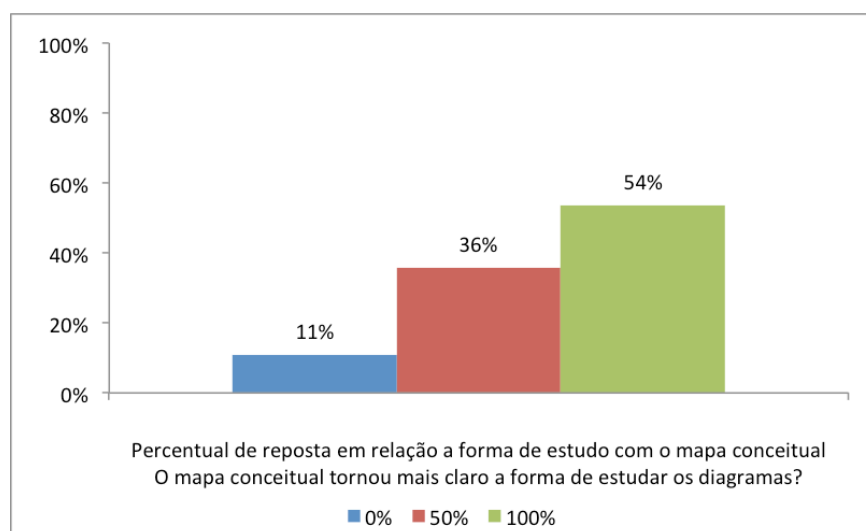


Figura 4.11: Histograma da questão 5 - Diagrama de Classe

O diagrama de classe possui muitos conceitos envolvidos, muito mais que o diagrama de caso de uso, sendo assim de acordo com a Figura 4.11 o mapa conceitual contribui na aprendizagem do diagrama.

Ao efetuar uma análise através dos questionamentos, em relação a contribuição para o ensino e aprendizagem quando utilizado as duas tecnologias, o resultado dos 27 alunos foi que, 7% não opinaram, 43% disseram que contribuiu “Parcialmente” e 50% dos alunos responderam “Totalmente” quando demonstrado as duas tecnologias a contribuição foi positiva, conforme demonstrado na Figura 4.12.

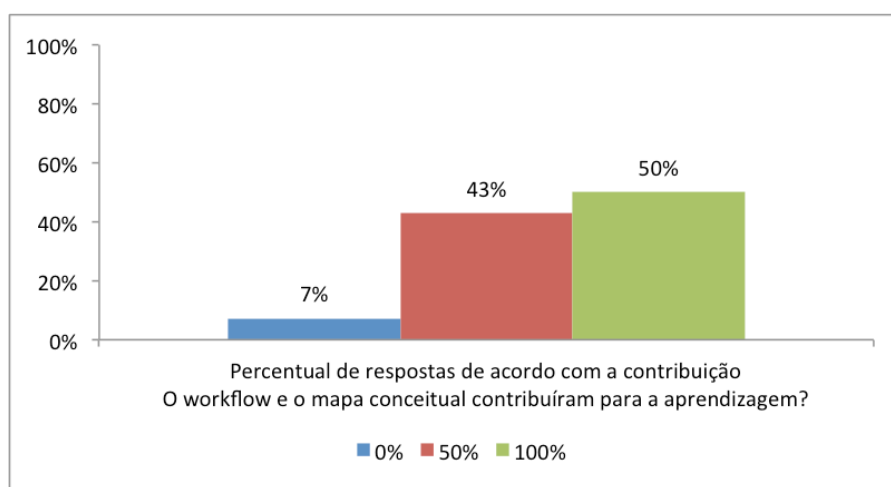


Figura 4.12: Histograma da questão 7 - Diagrama de Classe

No diagrama de classe, ao analisar as questões 3, 5 e 7, enfocando a utilização do *workflow* e dos mapas conceituais, as respostas foram, em todos os casos, satisfatórias, levando em consideração que em todas as 3 questões, os resultados foram positivos, para tanto se conclui que sua contribuição é significativa, conforme demonstrado nas Figuras 4.10, 4.11 e 4.12, respectivamente.

#### 4.1.4 Diagrama de Sequência

O diagrama de sequência é considerado pelos alunos, como um diagrama complexo.

A aula do diagrama de sequência foi ministrada para 44 alunos, e o resultado foi medido através do mesmo questionário atribuído aos demais diagramas.

Ao questionar os alunos que tiveram contato com a UML, quanto ao conhecimento agregado com o instrumento aplicado, de 44 alunos pesquisados, 82% responderam que tiveram 50% ou mais conhecimento agregado, sendo que para apenas 18% a resposta foi entre 20% a 40%, de acordo com o gráfico da Figura 4.13.

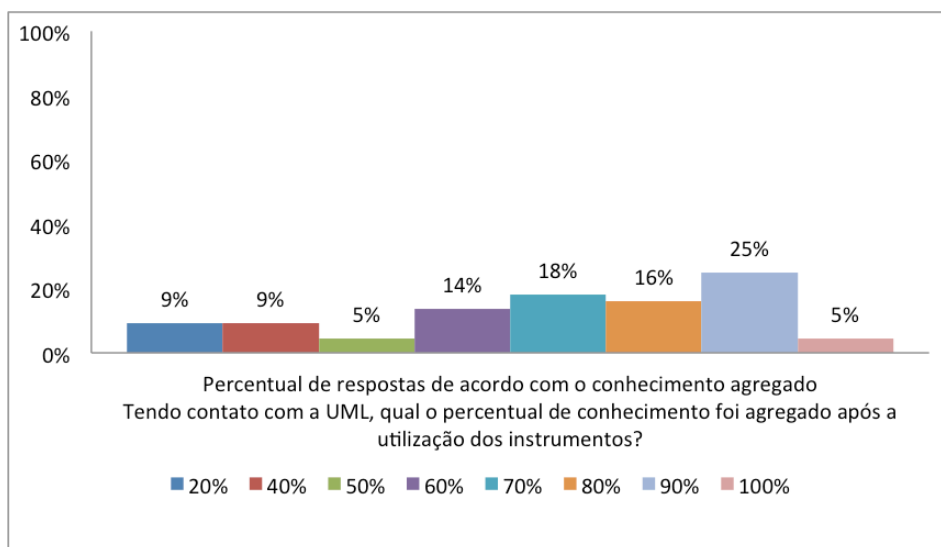


Figura 4.13: Histograma da questão 2 - Diagrama de Sequência

A Figura 4.13 demonstrou opiniões bastante equilibradas em relação ao conhecimento agregado quando já se conhecia o assunto em questão.

As instruções de trabalho complementam o *workflow*, onde é detalhado a atividade, os recursos utilizados, produtos resultantes entre outros itens.

Foi questionado os alunos se as instruções de trabalho juntamente com o *workflow* contribuíram para a aprendizagem. Dos alunos que responderam o questionamento, 5% responderam “Sem opinião”, 36% responderam “Parcialmente” e 59% responderam “Totalmente”.

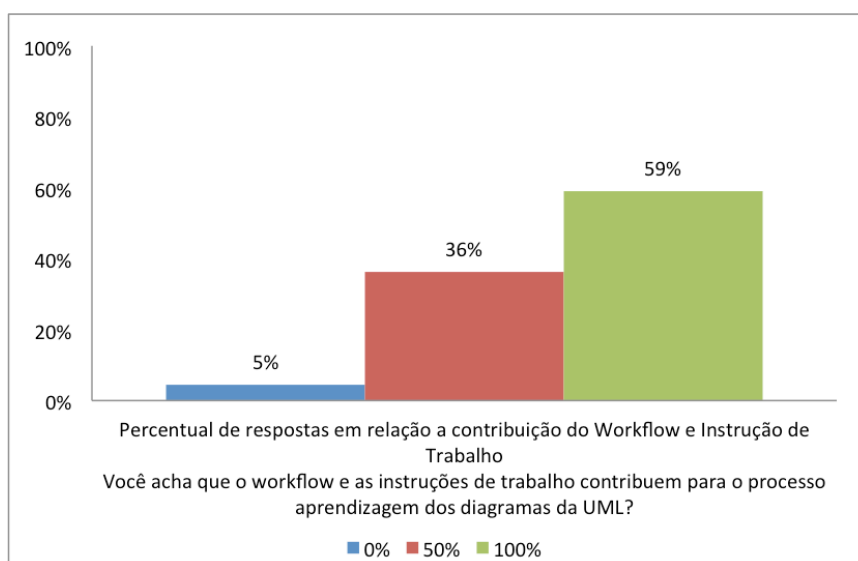


Figura 4.14: Histograma da questão 3 - Diagrama de Sequência

Para cada atividade do *workflow* foi elaborado uma instrução de trabalho, de forma descritiva e completa. Ao questionar em relação a contribuição da instrução de trabalho juntamente com o *workflow*, os alunos pesquisados consideraram de bastante importância, visto que somente 5% responderam que não sabiam opinar.

Com o objetivo de analisar a utilização do mapa conceitual para o estudo do diagrama de sequência, 9% dos alunos não opinaram, 34% responderam que a utilização do mapa conceitual auxiliou “Parcialmente” 57% responderam “Totalmente”, de acordo com a Figura 4.15.

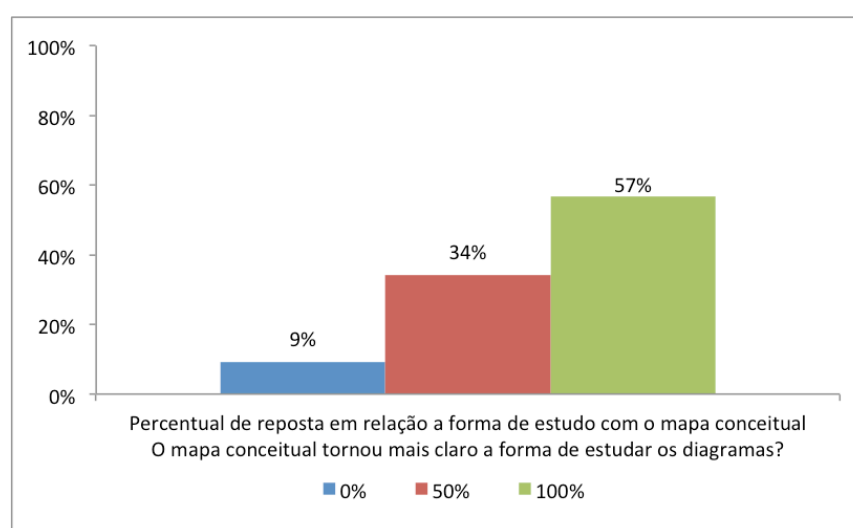


Figura 4.15: Histograma da questão 5 - Diagrama de Sequência

Quando analisado o conjunto, indagando se o mapa conceitual e o *workflow* contribuíram para a aprendizagem, 7% responderam “Sem Opinião”, 39% responderam “Parcialmente” e 55% responderam “Totalmente”, conforme demonstrado na Figura 4.16.

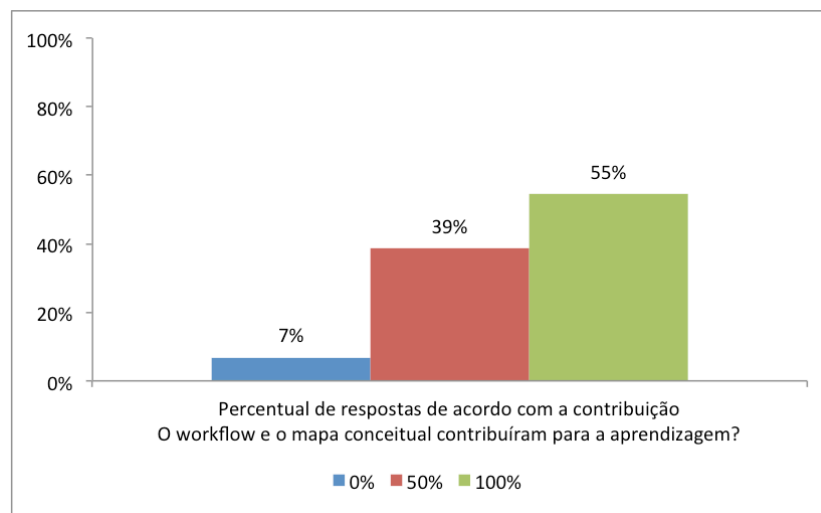


Figura 4.16: Histograma da questão 7 - Diagrama de Sequência

O estudo aplicado no diagrama de sequência apresentou-se de forma significativamente positiva, levando em consideração que dentre os 44 alunos que participaram das aulas, que as respostas foram positivas quando questionado em relação à utilização do *workflow* e dos mapas conceituais, para tanto se conclui que sua contribuição é significativa, conforme demonstrado nas Figuras 4.14, 4.15 e 4.16, respectivamente.

## 4.2 CONSIDERAÇÕES FINAIS

Tendo em vista todos os instrumentos aplicados em sala de aula e o levantamento dos resultados através dos questionários aplicados, foi concluído que os alunos e participantes do processo aprovaram o instrumento, visto que o *workflow* elaborado tornou-se completo e significativo com o mapa conceitual.

## 5 CONCLUSÕES E PESQUISAS FUTURAS

A educação é o processo de desenvolvimento de aptidões, de atitudes e de outras formas de conduta exigidas pela sociedade, um processo globalizado que visa à formação integral de uma pessoa, para o atendimento às necessidades e às aspirações de natureza pessoal e social.

Professores e alunos são os principais envolvidos no processo educacional, dessa forma, buscar maneiras de facilitar o processo de ensino e aprendizagem é relevante para auxiliar na formação dos novos profissionais.

Neste trabalho foi desenvolvido um processo de ensino aprendizagem utilizando mapas conceituais no ensino da UML possibilitando um melhor entendimento dos conceitos e aplicações por parte de alunos da área da computação.

*Workflows* foram criados para representar o processo permitindo estabelecer a ordem de execução das atividades e as condições em que cada atividade pode ser iniciada.

Os *workflows* foram, também, utilizados para demonstrar os passos a serem seguidos para a elaboração dos diagramas da UML. Os mapas conceituais apresentaram os conceitos relevantes de acordo com cada diagrama, tornando assim na perspectiva do aluno e do professor, um instrumento de fácil entendimento, onde o mapa conceitual complementa o *workflow*.

Para cada diagrama abordado, foi desenvolvido um ou mais *workflows*/mapa(s) conceitual(is). Por meio do *workflow* foi demonstrado o fluxo de atividades para o desenvolvimento de um dado diagrama, no qual as informações tramitaram entre os atores envolvidos, gerando artefatos relevantes para o entendimento. Em relação aos mapas conceituais, foram elaborados de forma gráfica e acompanhados da descrição dos conceitos e notação dos mesmos.

O instrumento foi aplicado para alunos ingressantes e concluintes de duas universidades. Alguns alunos já obtinham o conhecimento dos diagramas abordados e outros alunos não conheciam a UML. Para ambos os alunos, os mapas conceituais e o *workflow* auxiliaram no processo de ensino e aprendizagem. Para avaliar o aproveitamento do instrumento, após a aula, os alunos foram submetidos a um questionário. Os resultados foram positivos conforme foi apresentado na seção 4.1.

Conclui-se, também, que é possível utilizar o *workflow* e o mapa conceitual de forma colaborativa e simultânea agregando valores e contribuindo dessa forma para o processo de ensino e aprendizagem.

Com o resultado da utilização do *workflow* e dos mapas conceituais, foi possível elaborar os instrumentos de ensino e aprendizagem, buscando auxiliar o professor a ministrar as suas aulas didaticamente e aos alunos a estarem visualizando os conceitos relacionados ao assunto, antes demonstrado sem um processo de ensino aprendizagem.

Como contribuição da pesquisa, pode-se citar:

- aplicação da aprendizagem significativa por meio dos mapas conceituais;
- elaboração dos mapas conceituais para os diagramas propostos;
- elaboração dos *workflows* para os diagramas propostos;
- elaboração das instruções de trabalho para cada atividade do *workflow*;
- desenvolvimento do objeto de aprendizagem demonstrando aos alunos a utilização do *workflow* e mapas conceituais, onde os alunos podem navegar, verificar conceitos e seguir os passos necessários para a elaboração de um dado diagrama.

Como trabalhos futuros, o estudo pode ser estendido a outros diagramas da UML 2.x e também aplicar o estudo ao ensino à distância desenvolvendo, para isso, uma aplicação que permite a interatividade do aluno com a ferramenta, onde o aluno poderá acompanhar o processo pelo *workflow* buscando os conceitos relacionados nos mapas conceituais.

## 6 REFERÊNCIAS

- [1] PENDER, Tom. **UML, A Bíblia**, Rio de Janeiro: Elsevier, 2004.
- [2] LARMAN, C. **Utilizando UML e Padrões: uma Introdução a Análise e ao Projeto Orientados a Objetos**, Porto Alegre: Bookman, 2000.
- [3] BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivair. **UML: Guia do Usuário**, Rio de Janeiro: Elsevier, 2005.
- [4] QUATRINI, T. **Modelagem Visual com Rational Rose 2000 e UML**, Rio de Janeiro: Editora Ciência Moderna Ltda, 2001.
- [5] AUSUBEL, David et al. **Psicologia Educacional**, Rio de Janeiro: Interamericano, 1980.
- [6] BARBOSA, Márcio Lobo. **Mapas Conceituais na Avaliação da Aprendizagem significativa**, Disponível em: <http://www.sbf1.sbfisica.org.br/eventos/snef/xvi/cd/resumos/T0028-2.pdf> >. Acesso em Julho 2011.
- [7] MOREIRA, Marco Antonio. **Aprendizagem Significativa: a teoria de David Ausubel**, São Paulo: Centauro, 2001.
- [8] SIZILIO, Glaucia Regina M. A. Modelo de Autoria de Cursos de Ensino a Distância, **Revista Brasileira de Informática na Educação**, n. 8, Abril, 2001.
- [9] NOVAK, Joseph D. **Learning, Creating and using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations**, Laurence Erbaum Ass., 1998.
- [10] BARROS, Rodolfo Miranda de. **Um Estudo sobre o Poder das Metáforas e dos Recursos Multimídia no Processo de Ensino e Aprendizagem de Cálculo Diferencial e Integral**, Tese de Doutorado. Universidade Estadual de Campinas, Campinas, SP, 2008
- [11] TAROUCO, Liane; AMORETTI, Maria Suzana Marc. Mapas conceituais: modelagem colaborativa do conhecimento, **Revista Informática na Educação: Teoria & Prática**, v.3 n.1, PPGIE/UFRGS, Setembro, 2000.
- [12] PICHILIANI, M. C. **Usando a modelagem colaborativa no aprendizado da UML**, Disponível em <<http://www.br-ie.org/pub/index.php/wie/article/viewFile/906/892>>. Acesso em Junho 2006.
- [13] SANTOS, P. S.; MENEZES, C. S.; CURY, D.; NEVADO, R. A. **Um Ambiente para Acompanhamento da Aprendizagem baseado em Mapas Conceituais** In: Simpósio Brasileiro de Informática na Educação. Anais. Juiz de Fora, MG, 2005
- [14] LOPES, R. S., **Workflow Genético para Planejamento e Gerenciamento de Currículo em EAD**, Disponível em <<http://www.facom.ufu.br/posgrad/WD1/robson.pdf>>. Acesso em Junho de 2007.
- [15] ROBINSON, G., SCOPEL, M.. **Modelando Requisitos Especificados com Mapas conceituas através da UML-MC**, Manaus, Brasil, 2004.



- [16] ROLIM, Luiz Henrique Martins Lins. **Utilização de Mapas Conceituais em Engenharia de Software**: Projetando uma ferramenta case In: Encontro de Iniciação Científica e Pós-Graduação do ITA. Anais. São Paulo, SP, 2006.
- [17] LIMA, Adilson da Silva. **UML 2.0: do requisito à solução**, São Paulo: Erica, 3ª. Edição, 2008.
- [18] PRESSMAN, R. S. **Engenharia de Software**, São Paulo: Editora McGraw-Hill, 6a edição, 2006.
- [19] RATIONAL SOFTWARE CORPORATION, **Rational Unified Process**. Versão 2003.06.13. Copyright © 1987 – 2004. Rational Software Corporation 2004.
- [20] RATIONAL SOFTWARE CORPORATION, **Rational Unified Process**. Versão 7.1.1. Copyright © IBM Corporation 2000, 2007
- [21] MOREIRA, Marco Antonio. **Mapas conceituais e aprendizagem significativa**. Disponível em <<http://www.if.ufrgs.br/~moreira/mapasport.pdf>>. Acesso em 17 de dezembro de 2010
- [22] MOREIRA, Marco Antonio. **Aprendizagem Significativa: Um Conceito Subjacente**. Disponível em <<http://www.if.ufrgs.br/~moreira/apsigsubport.pdf>>. Acesso em 23 de dezembro de 2010.
- [23] SMOLE, Kátia Cristina Stocco. **Aprendizagem Significativa**. Disponível em <<http://www.fe.unb.br/pie/zAPRENDIZAGEM%20SIGNIFICATIVA.htm>>. Acesso em 27 de dezembro de 2010.
- [24] VALERIO, Marcus. **Teoria de Ausubel**. Disponível em <<http://www.xr.pro.br/monografias/ausubel.html>>. Acesso em 27 de dezembro de 2010.
- [25] MOREIRA, Marco Antonio. **Aprendizagem Significativa Crítica**. Disponível em <<http://www.if.ufrgs.br/~moreira/apsigcritport.pdf>>. Acesso em 27 de dezembro de 2010.
- [26] MENESTRINA, Tatiana Camiotto e Goudard Beatriz. Atualização e Revisão Pedagógica de Cálculo e Álgebra: **Concepções e Atitudes Inovadoras**. In: Associação Brasileira de Ensino de Engenharia, Congresso Brasileiro de Ensino de Engenharia, Rio de Janeiro, RJ, 2003.
- [27] BRIAN, R Gaines; MILDRED, L G Shaw. **Collaboration through Concept Maps**, Disponível em <<http://ksi.cpsc.ucalgary.ca/~gaines/reports/LW/CSCL95CM/CSCL95CM.pdf>>. Acesso em dezembro de 2004.
- [28] GUEDES, Gilleanes T. A. **UML 2.0: uma abordagem prática**, São Paulo: Novatec, 2009.
- [29] TELECKEN, Tiago Lopes. **Conceituais Um Estudo sobre modelos conceituais para ferramentas de definição de processos de workflow**. Porto Alegre, RS, 2004.
- [30] RUMBAUGH, James. **Modelagem e projetos baseado em objetos**. Rio de Janeiro: Campus, 1994.
- [31] MORTEN, C Meilgaard; GAIL, Vence Cville, Thomas Carr., **Sensory evaluation techniques**, chapter Statistical tables. CRC Press, 1999.

- [32] BARROS, R. M. **Alocação de Atividades em um Sistema de Gerência de Workflow**. Dissertação de Mestrado. CPGCC/UFRGS, Porto Alegre, 1997.
- [33] WORKFLOW MANAGEMENT COALITION. Terminology & Glossary. Disponível em <[http://www.wfmc.org/index.php?option=com\\_docman&Itemid=74](http://www.wfmc.org/index.php?option=com_docman&Itemid=74)>. Acesso em dezembro de 2010.
- [34] WORKFLOW MANAGEMENT COALITION. **The Workflow Reference Model**. Technical Report TC-1003. Disponível em <<http://www.wfmc.org>>. Acesso em janeiro de 2011.
- [35] JABLONSKI, S.; BUSSLER, C. **Workflow Management**. Modeling Concepts, Architecture and Implementation. England: Thomson Computer Press, 1996.
- [36] GEORGAKOPOULOS, D. et. al. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, **ACM Distributed and Parallel Database**, [S.I.], n. 3, p.119-153, 1995.
- [37] CASATI F. et. al. Conceptual Modeling of Workflows. In: OBJECT-ORIENTED AND ENTITY-RELATIONSHIP, 1995, Gold Coast, Australia. **Proceedings**. Gold Coast: [s.n.], 1995.
- [38] NICOLAO, Mariano. **Adaptação em um Curso na Web usando Workflow**. Tese de Doutorado. Universidade Federal do Rio Grande do Sul, Porto Alegre: RS, 2004
- [39] MEDINA-MORA, R. et al. The Action Workflow Approach to Workflow Management Technology. In: CSCW, 1992, [S.I.]. **Proceedings...** [S.l.:s.n.], 1992.
- [40] JOOSTEN, S. Trigger Modelling for Workflow Analysis. In: WORKFLOW MANAGEMENT CHALLENGES, PARADIGMS AND PRODUCTS, 1994, Viena, Austria. **Proceedings...** [S.I.:s.n.], 1994.
- [41] BIZAGI. *Process Modeler*. Disponível em <<http://www.bizagi.com>>. Acesso em Janeiro de 2010.

## **7 Trabalhos publicados pela Autora**

1. TANAKA, Simone ; BARROS, R. M. ; TANAKA, Sergio Akio. Utilização do Workflow e dos Mapas Conceituais no Processo de Ensino e Aprendizagem da UML. In: XXXVIII Congresso Brasileiro de Ensino de Engenharia, 2010, Fortaleza. XXXVIII Congresso Brasileiro de Ensino de Engenharia, 2010.
2. TANAKA, Simone ; BARROS, R. M. ; TANAKA, Sergio Akio. O Poder da Tecnologia de Workflow e dos Mapas Conceituais no Processo de Ensino e Aprendizagem da UML. In: FEES - Fórum de Educação em Engenharia de Software, 2010, Salvador. Simpósio Brasileiro de Engenharia de Software, 2010.
3. TANAKA, Simone ; BARROS, R. M. ; TANAKA, Sergio Akio ; EZEQUIEL, A. P.. O Poder da tecnologia de workflow e dos mapas conceituais no processo de ensino e aprendizagem da UML. In: XXXVI Conferência Latino-americana de Informática (XXXVI CLEI), 2010, Assunção. XXXVI Conferência Latino-americana de Informática, 2010.

## APÊNDICES

## APÊNDICE A

### Questionário

Nome (opcional): \_\_\_\_\_

Instituição: \_\_\_\_\_ Ano Graduação: \_\_\_\_\_

1. Você já teve algum contato com a UML?

☐ Sim

☐ Não

2. Tendo contato com a UML, que percentual de conhecimento foi agregado após a utilização dos instrumentos?

☐ 0% ☐ 10% ☐ 20% ☐ 30% ☐ 40%

☐ 50% ☐ 60% ☐ 70% ☐ 80% ☐ 90% ☐ 100%

3. Você acha que o *workflow* e as instruções de trabalho contribuem para o processo aprendizagem dos diagramas da UML?

☐ Totalmente

☐ Parcialmente

☐ Não

☐ Sem Opinião

4. Se Parcialmente, qual seria a porcentagem de contribuição?

☐ 10% ☐ 20% ☐ 30% ☐ 40% ☐ 50% ☐ 60% ☐ 70% ☐ 80% ☐ 90%

5. O mapa conceitual tornou mais clara a forma de estudar os diagramas?

☐ Totalmente

☐ Parcialmente

☐ Não

☐ Sem Opinião

6. Se Parcialmente, qual seria a porcentagem de contribuição?

☐ 10% ☐ 20% ☐ 30% ☐ 40% ☐ 50% ☐ 60% ☐ 70% ☐ 80% ☐ 90%

7. O workflow e o mapa conceitual contribuíram para a aprendizagem?

☐ Totalmente      ☐ Parcialmente      ☐ Não      ☐ Sem Opinião





8. Se Parcialmente, qual seria a porcentagem de contribuição?

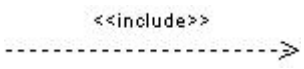
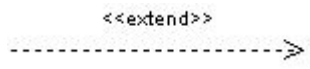
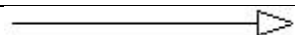
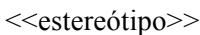
☐ 10%   ☐ 20%   ☐ 30%   ☐ 40%   ☐ 50%   ☐ 60%   ☐ 70%   ☐ 80%   ☐ 90%

## APÊNDICE B

### Definição dos Conceitos do Mapa Conceitual

Definição dos Conceitos do Mapa Conceitual (Diagrama de Caso de Uso)


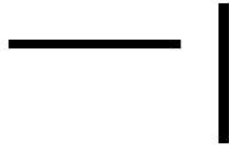
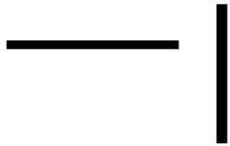
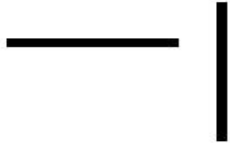


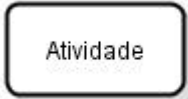
Nome		Definição	Notação
Caso de Uso		São utilizados para capturar os requisitos do sistema, ou seja, referem-se aos serviços, tarefas ou funcionalidades.	
Ator		Representa os papéis desempenhados pelos diversos usuários que poderão utilizar, de alguma maneira, os serviços e funções do sistema.	
Relacionamento		Interação entre o ator e o Caso de Uso.	Não há.
Relacionamento	Associação	O relacionamento de associação representa a informação de quais atores estão associados a que casos de uso, podendo ser unidirecional ou bidirecional.	Não há.
	Associação	Associação unidirecional	Indica o sentido em que as informações trafegam. 
		Associação bidirecional	As informações são transmitidas nas duas direções. 
	Dependência	Este relacionamento, como o próprio nome diz,	Não há.

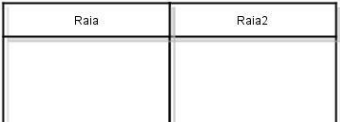

		identifica certo grau de dependência de um Caso de Uso em relação à outra. O relacionamento de dependência é apresentado por uma linha tracejada entre Caso de Uso.		
	Dependência	<i>Include</i>	Costuma ser utilizada quando existe um cenário comum a mais de um Caso de Uso. Indica a obrigatoriedade da execução do Caso de Uso incluído.	
		<i>Extend</i>	São utilizados para descrever cenários opcionais de um Caso de Uso. Representa eventos que não ocorrem sempre.	
	Generalização	Um relacionamento de um Caso de Uso-filho para Caso de Uso-pai.		
Estereótipo		Permite a extensibilidade aos componentes ou associação da UML. Como exemplo, temos o <i>include</i> e o <i>extend</i> .		
Requisitos		São definidos como condição ou capacidade com a qual o sistema deve estar de acordo, podendo ser funcional ou não-funcional.	Não há.	



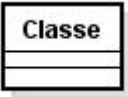
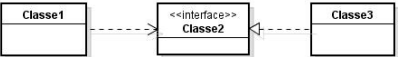

Requisito	Requisito Funcional	É definido como uma condição ou capacidade com a qual o sistema deve estar de acordo.	Não há.
	Requisito não-funcional	Descreve os atributos do sistema ou atributos do ambiente de sistema.	Não há.
Especificação de Caso de Uso		Descreve com linguagem simples as informações referentes ao Caso de Uso, quais atores interagem, os passos a serem executados, entre outros.	Verificar o Anexo.




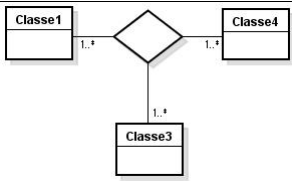


## Definição dos Conceitos do Mapa Conceitual (Diagrama de Atividades)

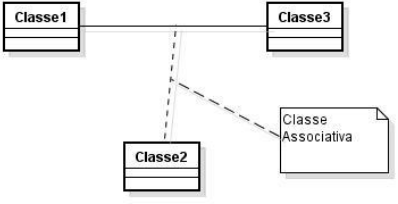
Nome		Definição	Notação
Início e Fim		Utilizado para delimitar o início e término de um diagrama de Atividades. O início é representado por um círculo preenchido, e o fim, por um círculo preenchido dentro de um círculo vazio.	
Condição de Guarda		É uma condição que pode ser atribuída a uma transição para restringir a transição entre as atividades. A condição de guarda pode ser representada entre colchetes.	[Condição de Guarda]
Concorrência		Descreve várias <i>threads</i> ou processos sendo executados simultaneamente.	
Concorrência	Bifurcação	Mostra uma transição iniciando várias transições.	
	União	Mostra várias transições terminando em uma nova transição assumindo.	
Decisões		Utilizado para representar uma escolha entre dois ou mais fluxos, sendo que um dos fluxos será escolhido [28].	
Transição		Também conhecido como fluxo de controle, é um conector que liga dois nós e é representado por uma linha com uma seta.	
Atividades		É uma etapa de um processo em que algum trabalho está sendo realizado.	

Raias de natação	Servem para particionar em grupos os estados de atividades de um diagrama de atividades, cada grupo representando a organização de negócios responsável por essas atividades.	
Ponto de <i>merge</i>	Local onde dois caminhos se juntam e continuam como um.	

## Definição dos Conceitos do Mapa Conceitual (Diagrama de Classe)



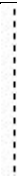

Nome		Definição	
Classe		Representa uma categoria, e os objetos são os membros ou exemplos dessa categoria. Em geral, uma classe tem atributos e métodos, mas é possível encontrar classes que contenham apenas uma dessas características ou mesmo nenhuma delas, como no caso de classes abstratas. Tais classes podem ser de dois tipos: abstratas e concretas.	
Multiplicidade		Especifica a quantidade de valores que podem estar associados a um elemento do modelo.	<b>[0..5] ou 0..5</b>
Relacionamento		As classes costumam ter relacionamentos entre si, chamados associações, que permitem que elas compartilhem informações entre si e colaborem para a execução dos processos executados pelo sistema [28].	
Relacionamento	Realização (Interface)	É um tipo de relacionamento especial que mistura características dos relacionamentos de generalização e dependência, sendo usada para identificar classes responsáveis por executar funções para outras classes [28].	
	Dependência	É o relacionamento, como o próprio nome diz, identifica certo grau de dependência de uma classe em relação à outra [28].	
	Generalização	O objetivo dessa associação é representar a ocorrência de herança entre as classes, identificando as	

Associação		superclasses, chamadas gerais e subclasses, chamadas especializadas, demonstrando a hierarquia entre as classes e possivelmente métodos polimórficos nas classes especializadas [28].	
	Associação	Uma associação descreve um vínculo que ocorre normalmente entre os objetos de uma ou mais classes [28].	
	Unidirecional	Relacionamento de um objeto de uma classe com objetos da mesma classe [28].	
	Bidirecional	Relacionamento entre objetos de duas classes distintas [28].	
	Ternária	São associações que conectam objetos de mais de duas classes. São representadas por um losango para onde convergem todas as ligações da associação [28].	
	Agregação	É um tipo especial de associação onde se tenta demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe (chamados objetos-parte). Esse tipo de associação tenta demonstrar uma relação todo/parte entre os objetos associados [28].	
	Composição	É um tipo de composição da qual constitui-se em uma variação da agregação, onde é apresentado um vínculo mais forte entre os objetos-todo e os objetos-parte, procurando demonstrar que os objetos-parte têm de estar associados a um único objeto-todo[28].	

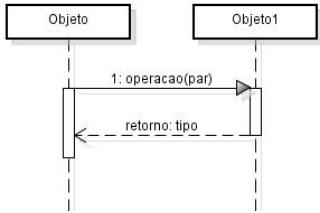
		Classe Associativa	São aquelas produzidas quando da ocorrência de associações que tenham multiplicidade muitos (*) em todas as suas extremidades [28].	
Estereótipo			Permite a extensibilidade aos componentes ou associação da UML.	<b>&lt;&lt;entity&gt;&gt;</b> <b>&lt;&lt;boundary&gt;&gt;</b> <b>&lt;&lt;control&gt;&gt;</b> <b>&lt;&lt;enumeration&gt;&gt;</b>
Atributos			Representam as características de uma classe, ou seja, as peculiaridades que costumam variar de um objeto para outro [28].	
Atributo	Derivado	é o atributo do qual seus valores são representados por algum tipo de calculo. Nestes casos o atributo é representado por uma / na frente do atributo [28].	<b>/atributo</b>	
	Estático	São atributos cujos valores são idênticos para todos os objetos de uma classe, ou seja, é um atributo pertencente à classe propriamente dita. Sua identificação se dá através do nome da variável sublinhado [28].	<b><u>atributo</u></b>	
	Tipos	O tipo de atributo identifica um classificador que explica a espécie de informação que pode ser armazenada no atributo [1].	<b>int,</b> <b>boolean,</b> <b>varchar,</b> <b>double</b>	
Visibilidade			É utilizada para indicar o nível de acessibilidade de um determinado atributo ou método, sendo representada à esquerda destes, existindo basicamente quatro modos de visibilidade: público,	

		protegido, privado e pacote [28].	
Visibilidade	Publica	Determina que o atributo ou método pode ser utilizado por qualquer objeto [Gilleanes].	<b>+ atributo3</b>
	Privada	Somente os objetos da classe detentora do atributo ou método poderão enxergá-lo [Gilleanes].	<b>- atributo1</b>
	Pacote	Determina que o atributo ou método é visível por qualquer objeto dentro do pacote. Somente elementos que fazem parte de um pacote podem ter essa visibilidade. Nenhum elemento fora do pacote poderá ter acesso a um atributo ou método com essa visibilidade.	<b>~ atributo4</b>
	Protegida	Determina que além dos objetos da classe detentora do atributo ou método também os objetos de suas subclasses poderão ter acesso ao mesmo [Gilleanes].	<b># atributo2</b>
Operação		Também conhecido como método ou comportamento da qual representa uma atividade que um objeto de uma classe pode executar [28].	<b>operação() : int</b> <b>operação() : boolean</b>


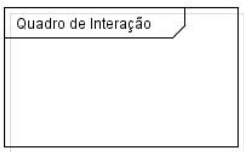
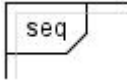
## Definição dos Conceitos do Mapa Conceitual (Diagrama de Sequencia)


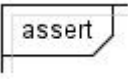
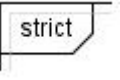
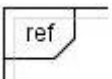
Nome	Definição	Notação
Ator	Representa os papéis desempenhados pelos diversos usuários que poderão utilizar, de alguma maneira, os serviços e funções do sistema. Os atores neste diagrama são instâncias dos atores declarados no diagrama de Casos de Uso, representam entidades externas que iniciam processos.	
Objeto/Classe	Objetos são membros ou exemplos de uma determinada categoria, que é representada por uma classe.	
<i>Lifelines</i>	Um <i>lifeline</i> é um participante individual em uma interação. Na maioria das vezes um <i>lifeline</i> irá se referir a uma instância de uma classe [28].	Não há
Linha de Vida	A linha de vida representa o tempo em que um objeto ( <i>lifeline</i> ) existe durante um processo. As linhas de vida são representadas por linhas finas verticais tracejadas, partindo do retângulo que representa o objeto [1].	
Foco de Controle	Indica os períodos em que um determinado objeto está participando ativamente do processo, ou seja, identifica os momentos em que um objeto está executando um ou mais métodos utilizados em um processo	

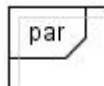
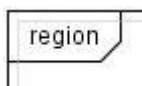

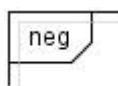



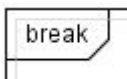
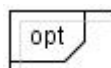
		específico. Os focos de controle são representados dentro da linha de vida de um objeto [28].	
Mensagem	Mensagem	As mensagens são utilizadas para demonstrar a ocorrência de eventos, que normalmente forçam a chamada de um método em alguns dos objetos envolvidos no processo. Pode ocorrer, no entanto, de uma mensagem representar a comunicação entre dois atores, nesse caso, não disparando métodos [28].	Não há
	Condição de Guarda	Uma expressão de guarda determina se uma mensagem pode ser executada. A guarda é expressa como uma restrição Booleana sobre os valores disponíveis para testar no momento da mensagem [1].	<b>‘[‘ guarda ‘]’</b>
	Retorno	Esse tipo de mensagem identifica a resposta a uma mensagem para o objeto ou ator que a chamou. Uma mensagem de retorno pode retornar informações específicas do método chamado ou apenas um valor indicando se o método foi executado com sucesso ou não. As mensagens de retorno são representadas por uma linha tracejada, contendo uma seta fina que aponta para o objeto que recebe o resultado do método chamado [1].	 <pre> sequenceDiagram     participant Objeto     participant Objeto1     Objeto-&gt;&gt;Objeto1: 1: operacao(par)     Objeto1--&gt;&gt;Objeto: retorno: tipo </pre>
	Operação	Também conhecidos como métodos, ou comportamentos.	<b>operacao()</b>

		Uma operação representa uma atividade que um objeto de uma classe pode executar [28].	
	Parâmetros	É uma lista ordenada dos atributos que, juntos, definem a entrada para uma operação. A lista de parâmetros é opcional, pois uma operação não precisa ter parâmetros. Segue ao lado um método contendo dois parâmetros representados pela abreviação <i>par</i> [1].	<b>Operação (par: String, par2: int)</b>
	Recursão	Um objeto também poderia ter de chamar uma mensagem recursivamente, ou seja, chamar a mesma mensagem de dentro da mensagem [1].	<pre> sequenceDiagram     participant Start     participant Objeto     participant Objeto1     Start-&gt;&gt;Objeto: 1:     activate Objeto     Objeto-&gt;&gt;Objeto1: 2: operacao(par)     activate Objeto1     Objeto1-&gt;&gt;Objeto1: 2.1: operacao1(par)     activate Objeto1     Objeto1--&gt;&gt;Objeto:      deactivate Objeto1     Objeto--&gt;&gt;Start:      deactivate Objeto </pre>
	Auto-referência	São mensagens que um objeto envia para si mesmo. No caso de auto-referência, uma mensagem parte da linha de vida do objeto e atinge a linha de vida do próprio objeto [28].	<pre> sequenceDiagram     participant Objeto     Objeto-&gt;&gt;Objeto: 1: operacao()     activate Objeto     Objeto--&gt;&gt;Objeto:      deactivate Objeto </pre>
	Síncrono	Uma mensagem síncrona considera que um retorno é necessário, de modo que o transmissor espera pelo retorno antes de prosseguir com qualquer outra atividade [1].	<pre> sequenceDiagram     participant A     participant B     A-&gt;&gt;B: 1: Message1(par)     activate B     B--&gt;&gt;A: return: tipo     deactivate B </pre>
	Assíncrono	Uma mensagem assíncrona diz algo sobre as responsabilidades do transmissor e do receptor. O emissor é responsável apenas por levar a mensagem ao receptor [1].	<pre> sequenceDiagram     participant A     participant B     A-&gt;&gt;B: 1: Message1(par)     activate B     B--&gt;&gt;A:      deactivate B </pre>

		<i>Balking</i>	Significa que, se o destinatário da mensagem não estiver imediatamente pronto para aceitar a mensagem, o remetente aborta a mensagem e continua o processamento.	
Quadro de Interação		Quadro de Interação	Seu objetivo é fornecer uma visão geral do controle de fluxo, oferecendo os meios de isolar um conjunto de interações reutilizável. Existem basicamente dois tipos de quadros: quadros de interação, que contêm qualquer tipo de diagrama de interação da UML, e quadros de ocorrência de interação, que normalmente fazem referência a um diagrama de interação, mas não apresentam seu detalhamento [28].	
	Tag	Seq	O operador de interação seq (sequência fraca) força as interações a seguirem uma certa ordem. A ordem é baseada no posicionamento dos fragmentos de interação dentro do fragmento combinado da mesma maneira como as mensagens de eventos são ordenadas por seu posicionamento na linha de vida vertical. Se dois eventos ocorrem na mesma linha de vida, o evento na interação mais alta é executado em primeiro lugar. [1]	
		<i>Ignore/Consider</i>	O operador de interação <i>ignore</i> identifica explicitamente as mensagens às quais o fragmento	

			<p>de interação não deve responder. O operador de interação <i>consider</i> identifica explicitamente as mensagens que devem ser tratadas, e é o equivalente a rotular cada mensagem alternada como sendo ignorada. Os dois operadores melhoram a facilidade de modelagem, porque permitem que o modelador focalize o comportamento mais relevante, sem especificar cada exceção [1].</p>	
		<i>Assert</i>	<p>O operador de interação <i>assert</i> (asserção) identifica um operando como única continuação de eventos válida em um fragmento da interação. Os cenários alternativos normalmente são modelados por meio de outros diagramas de sequência [1].</p>	
		<i>Strict</i>	<p>O operador de interação <i>strict</i> (sequência estrita) define a ordem de execução para um conjunto de fragmentos de interação. A ordenação é limitada às interações no primeiro nível de um fragmento de interação aninhado. O efeito de <i>strict</i> é forçar o término de uma interação antes de passar à próxima interação [1].</p>	
		Ref	<p>O termo “ref” é colocado na área de nomes superior esquerda do quadro. Isso alerta o leitor de que o conteúdo do quadro é definido em outro lugar, ou seja, no diagrama cujo nome está na área</p>	

		de conteúdo [1].	
	Par	O operador de interação Par (paralela) admite a execução paralela de um conjunto de fragmentos de interação. A integridade de cada fragmento de interação é preservada, enquanto permite que a execução exata dos eventos dentro de cada fragmento de interação seja intercalada de qualquer maneira admitida pelo ambiente de processamento [1].	
	Region	O operador de interação Region (região crítica) identifica uma interação que tem precedência sobre todos os outros fragmentos de interação dentro do contexto do fragmento combinado [1].	
	Alt	O operador de interação Alt (alternativas) identifica um conjunto de comportamentos que a interação pode escolher, com base nos critérios especificados. Porém, como acontece com qualquer operando, o operando selecionado na estrutura Alt só é executado, se a condição de guarda for avaliada como verdadeira [1].	
	Neg	O operador de interação Neg (negação) é usado para identificar interações que não deverão acontecer. Essas interações são descritas explicitamente como inválidas. O operador Neg pode oferecer uma abreviação	

			conveniente, quando qualquer quantidade de opções for válida, mas somente uma ou uma pequena quantidade de opções não for permitida – pode ser mais fácil e menos demorado identificar as exceções [1].	
		<i>Loop</i>	O operador de interação <i>loop</i> indica que o fragmento da interação será executado repetidamente. O número de vezes que ele é executado é determinado pelos parâmetros <i>minint</i> e <i>maxint</i> do operador [1].	
		<i>Break</i>	Oferece um mecanismo semelhante à sintaxe de interrupção em muitas linguagens de programação. No decorrer da execução de uma interação, se a guarda da interrupção for satisfeita, então, a interação que contém abandona sua execução normal e, em vez disso, realiza a cláusula especificada pelo fragmento <i>break</i> [1].	
		<i>Opt</i>	Representa um comportamento que pode ou não ser usado como parte da interação. Para ser usado, a condição de guarda precisa ser satisfeita. Se a condição de guarda falhar, o comportamento é simplesmente pulado. O modelo para um fragmento combinado <i>Opt</i> se parece com um <i>Alt</i> que oferece apenas uma interação [1].	

**ANEXO**

**ANEXO**  
**ESPECIFICAÇÃO DE CASO DE USO [20]**

Versão <1.0>

*[Observação: O template a seguir é fornecido para uso com o Rational Unified Process. O texto em azul exibido entre colchetes e em itálico (style=InfoBlue) foi incluído para orientar o autor e deve ser excluído antes da publicação do documento. Qualquer parágrafo inserido após esse estilo será definido automaticamente como normal (style=BodyText).]*

*[Para personalizar campos automáticos no Microsoft Word (que exibem um plano de fundo cinza quando selecionados), selecione File>Properties e substitua o conteúdo dos campos Title, Subject e Company pelas informações adequadas a esse documento. Depois de fechar a caixa de diálogo, para atualizar os campos automáticos no documento inteiro, selecione Edit>SelectAll (ou Ctrl-A) e pressione F9 ou simplesmente clique no campo e pressione F9. Isso deve ser feito separadamente para Cabeçalhos e Rodapés. Alt-F9 alterna entre a exibição de nomes de campos e a do conteúdo de campos. Consulte a ajuda do Word para obter mais informações sobre como trabalhar com campos.]*



## Histórico da Revisão

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
<dd/mm/aa>	<x.x>	<detalhes>	<nome>

Índice Analítico	
1.Nome do Caso de Uso	114
1.1Breve Descrição	114
2. Fluxo de Eventos	114
2.1Fluxo Básico	114
2.2 Fluxos Alternativos	115
2.2.1<Primeiro Fluxo Alternativo >	115
2.2.2< Segundo Fluxo Alternativo >	115
3.Requisitos Especiais	115
3.1<Primeiro Requisito Especial >	116
4.Precondições	116
4.1<Precondição Um>	116
5.Pós-condições	116
5.1<Pós-condição Um >	116
6.Pontos de Extensão	116
6.1<Nome do Ponto de Extensão>	116

Especificação de Caso de Uso: <Nome do Caso de Uso>

*[O template a seguir é fornecido para uma Especificação de Caso de Uso, que contém as propriedades textuais do Caso de Uso. Este documento é usado com uma ferramenta de gerenciamento de requisitos, como o Rational Requisite Pro, para especificar e marcar os requisitos contidos nas propriedades do Caso de Uso.*

*Os diagramas de Caso de Uso podem ser desenvolvidos em uma ferramenta de modelagem visual, como o Rational Rose. Um relatório de Caso de Uso, com todas as propriedades, pode ser gerado com o Rational SoDA. Para obter mais informações, consulte os mentores de ferramentas do Rational Unified Process.]*

Nome do Caso de Uso

Breve Descrição

*[A descrição relata brevemente o papel e a finalidade do Caso de Uso. Um único parágrafo será suficiente para essa descrição.]*

Fluxo de Eventos

Fluxo Básico

*[Este Caso de Uso é iniciado quando o ator pratica alguma ação. Os casos de uso sempre são iniciados por atores. O Caso de Uso descreve o que o ator faz e o que o sistema faz em resposta. Ele deve ser elaborado como um diálogo entre o ator e o sistema.*

*O Caso de Uso descreve o que ocorre no sistema, mas não como ou por quê. Se forem trocadas informações, seja específico no que diz respeito ao conteúdo que é passado e retornado. Por exemplo, não é muito esclarecedor dizer que o ator fornece informações do cliente. É melhor dizer que ele fornece o nome e o endereço do cliente. É útil fazer uso de um Glossário de Termos para manter a complexidade do Caso de Uso sob controle — poderá ser conveniente definir termos como, por exemplo, informações do cliente neste glossário, a fim de evitar que o Caso de Uso fique repleto de detalhes.*

*As alternativas simples poderão ser apresentadas no texto do Caso de Uso. Se forem necessárias apenas algumas frases para descrever o que acontece quando há uma alternativa, faça essa descrição diretamente na seção **Fluxo de Eventos**. Se o fluxo*

*alternativo for mais complexo, use uma seção separada para descrevê-lo. Por exemplo, uma subseção **Fluxo Alternativo** explica como descrever alternativas mais complexas. Às vezes, uma figura vale por mil palavras, embora não haja nada que possa substituir uma redação clara e organizada. Se for mais esclarecedor, sinta-se à vontade para colar representações gráficas de interfaces do usuário, fluxos de processo ou outras imagens no Caso de Uso. Se um fluxograma for útil para apresentar um processo complexo de decisões, utilize-o sem dúvida nenhuma! Assim como no caso de comportamentos dependentes de estado, um diagrama de transição de estado geralmente esclarece o comportamento de um sistema muito mais do que páginas e páginas de texto. Use o meio de apresentação certo para o problema, mas procure evitar o uso de terminologia, notações ou imagens que o público possa não entender. Lembre-se de que sua finalidade é esclarecer e não obscurecer.]*

## Fluxos Alternativos

### < Primeiro Fluxo Alternativo >

*[As alternativas mais complexas são descritas em uma seção separada, mencionada na subseção **Fluxo Básico** da seção **Fluxo de Eventos**. Pense nas subseções **Fluxo Alternativo** como comportamentos alternativos — cada fluxo alternativo representa um comportamento alternativo geralmente devido a exceções que ocorrem no fluxo principal. O tamanho desses fluxos poderá ser tão extenso quanto o necessário para descrever os eventos associados ao comportamento alternativo. Quando um fluxo alternativo termina, os eventos do principal fluxo de eventos são retomados, a menos que seja especificado algo em contrário.]*

### < Um Subfluxo Alternativo >

*[Os fluxos alternativos, por sua vez, podem ser divididos em subseções, se isso contribuir para maior clareza.]*

### < Segundo Fluxo Alternativo >

*[Pode haver, e muito provavelmente haverá, uma série de fluxos alternativos em um Caso de Uso. Mantenha cada fluxo alternativo separado para aumentar a clareza. O uso de fluxos alternativos melhora a legibilidade do Caso de Uso e evita que os casos de uso sejam decompostos em hierarquias de casos de uso. Lembre-se de que os casos de uso são apenas descrições textuais e que sua finalidade principal é documentar o comportamento de um sistema de maneira clara, concisa e compreensível.]*

## Requisitos Especiais

*[Normalmente, um requisito especial é um requisito não-funcional específico de um Caso de Uso mas que não é especificado de maneira fácil ou natural no texto do fluxo de eventos do Caso de Uso. Entre os exemplos de requisitos especiais estão incluídos requisitos legais e reguladores, padrões de aplicativos e atributos de qualidade do sistema a ser criado, incluindo requisitos de usabilidade, confiabilidade, desempenho ou suportabilidade. Além disso, outros requisitos — como ambientes e sistemas operacionais, requisitos de compatibilidade e restrições de design — deverão ser capturados nesta seção.]*

< Primeiro Requisito Especial >

#### Precondições

*[Uma precondição de um Caso de Uso é o estado do sistema que deve estar presente antes de um Caso de Uso ser realizado.]*

< Precondição Um >

#### Pós-condições

*[Uma pós-condição de um Caso de Uso é uma lista dos possíveis estados em que o sistema poderá se encontrar imediatamente depois do término de um Caso de Uso.]*

< Pós-condição Um >

#### Pontos de Extensão

*[Pontos de extensão do Caso de Uso.]*

<Nome do Ponto de Extensão>

*[Definição da localização do ponto de extensão no fluxo de eventos.]*